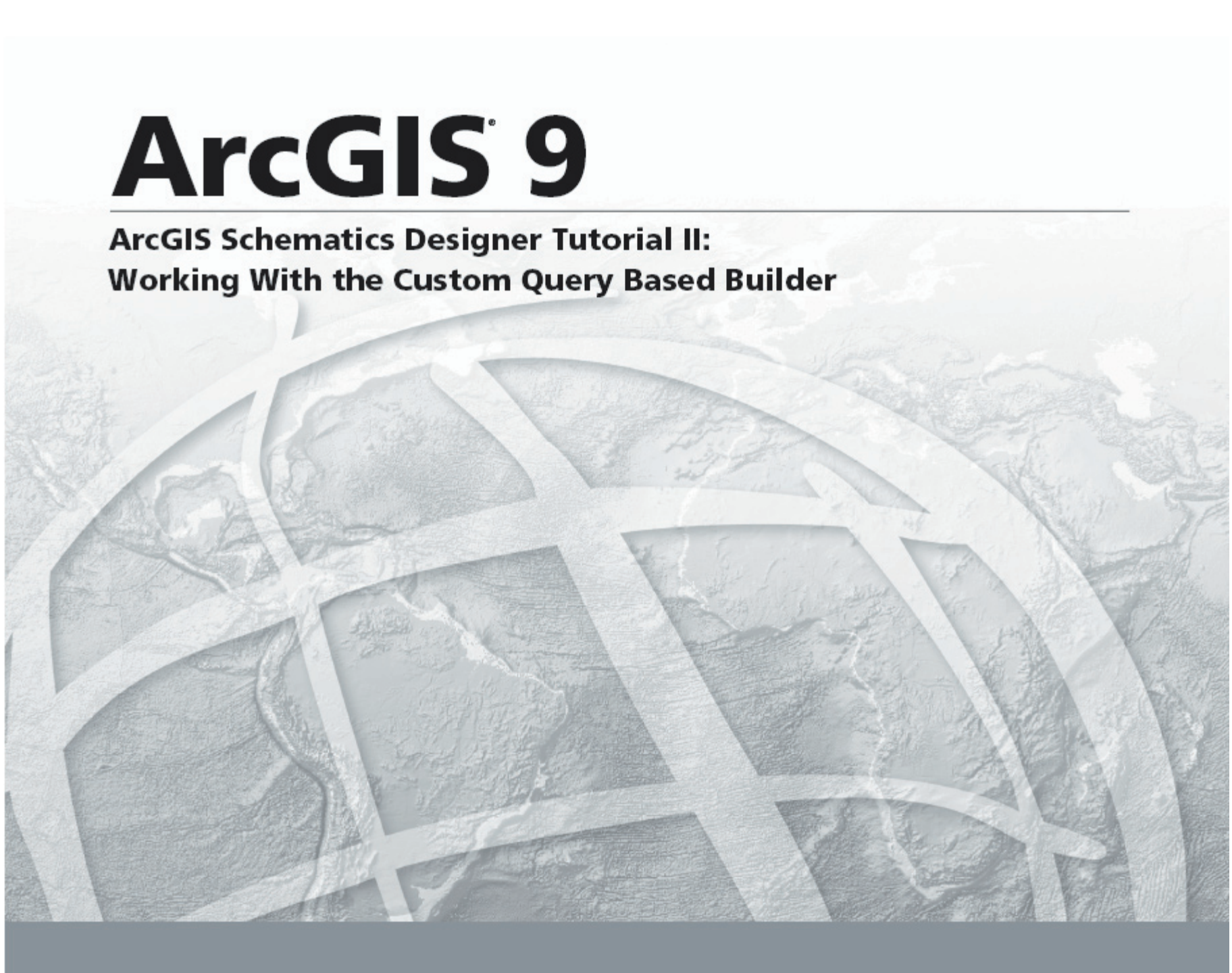
ArcGIS® 9

ArcGIS Schematics Designer Tutorial II: Working With the Custom Query Based Builder



Copyright © 1999–2006 ESRI.

All rights reserved.

Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and the copyright laws of the given countries of origin and applicable international laws, treaties, and/or 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 Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

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

CONTRIBUTING WRITER

Anne-Yvonne Blin

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 Alternates 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, ArcView, SDE, and the ESRI globe logo, ArcInfo, ArcSDE, ArcCatalog, ArcMap, ArcToolbox, ArcStorm, ArcEditor, ArcGIS, ArcObjects, StreetMap, the ESRI Press logo, and GIS by ESRI, and www.esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions.

The names of other companies and products herein are trademarks or registered trademarks of their respective trademark owners.

Contents

Introducing the Schematics Designer tutorial II

Welcome to the ArcGIS Schematics Designer tutorial II.
This tutorial has been designed to assist you in your ArcGIS Schematics project conception. By specifying diagrams that will work with the Custom Query-Based Builder, it will help you understand more about how to customize your schematic project—the entry point to access your schematic definition environment. This schematic project is a set of definitions and configurations, which references all primary schematic components managed by your schematics application.

Introducing the sample database's content

A summary of the Tutorial II database

When the ArcGIS Schematics tutorials were installed on your computer, the C:\arcgis\ArcTutor\Schematics\Designer folder should have been created. This directory contains the Schematics_Designer_Tutorial database in which this tutorial is based.

This sample database contains an electrical network. The data describing this network is organized into four tables: Station, Feeder, LV_Line, and HV_Line. The LV_Line and HV_Line tables list all the feature edges (links) of the network. The Station and Feeder tables contain all the network feature junctions (nodes).

The complete network is 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. In the latter case, the link that connects the station to the feeder connects the low-voltage part of the network to the high-voltage part.

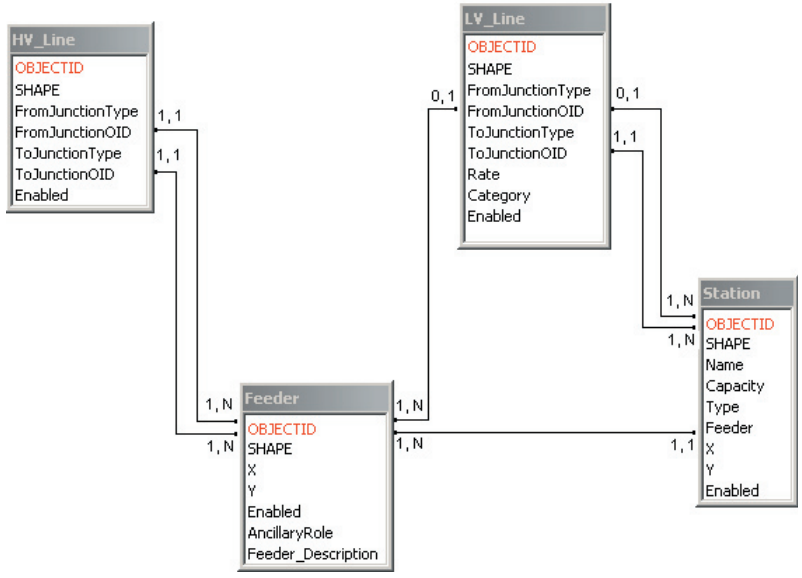
Specifications for the tutorial schematic project

The purpose is to graphically display the electrical network data contained in the Schematics_Designer_Tutorial database. The final schematic project will be organized into two main types of schematic diagrams that will both work with the Custom Query-Based Builder:

- The first will contain the complete network (low-voltage and high-voltage parts).
- The second will allow the display of network data related to a single feeder subnetwork (that is, related to a given Low-voltage part of the network, each one is managed by one feeder node).

Tutorial database's table of contents

The data model can be presented as follows:



The Station table contains six fields:

- OBJECTID: Station identifier (number)
- SHAPE: ESRI Binary Large Object (BLOB) field
- Name: Station name (string)
- Capacity: Station capacity (number)
- Type: Station type (a variable coded as A, B, C, D, E, or F)
- Feeder: Feeder OBJECTID to which the current station belongs (a variable coded as 1, 2, or 3)

The LV_Line table contains eight fields:

- OBJECTID: Low-voltage line identifier (number)
- SHAPE: ESRI BLOB field
- FromJunctionType: Low-voltage line's origin node type (a variable coded as station or feeder depending on the type of the origin node)
- FromJunctionOID: Low-voltage line's origin identifier (number)
- ToJunctionType: Low-voltage line's extremity node type (a variable coded as station or feeder depending on the type of the extremity node)
- ToJunctionOID: Low-voltage line's extremity identifier (number)
- Rate: Ratio value associated with the Low-voltage line (number)
- Category: Low-voltage line category (a variable coded as S, M, or B)

The Feeder table contains two fields:

- OBJECTID: Feeder node identifier (number)
- SHAPE: ESRI BLOB field

The HV_Line table contains six fields:

- OBJECTID: High-voltage line identifier (number)
- SHAPE: ESRI BLOB field
- FromJunctionType: High-voltage line's origin node type (a variable coded as Station or Feeder depending on the type of the origin node)
- FromJunctionOID: High-voltage line's origin identifier (number)
- ToJunctionType: High-voltage line's extremity node type (a variable coded as station or feeder depending on the type of the extremity node)
- ToJunctionOID: High-voltage line's extremity identifier (number)

Introducing the tutorial main steps

In this tutorial, the schematic project conception is organized into eight exercises:

- The purpose of this first exercise will be to create the schematic dataset that will contain all the schematic data related to the schematic project you will build and to edit the related schematic project within Schematics Designer.
- During the second exercise, you will become familiar with the schematic project's main parameters and you will specify default values for these parameters.
- The purpose of the third exercise will be to create all schematic components required to display all network data contained in the sample database in a unique schematic diagram managed by the Custom Query-Based Builder.
- During the fourth exercise, you will learn how to animate and document network elements displayed in schematic diagrams according to some database field values.
- In the fifth exercise, you will examine the low-voltage part of the network. The purpose will be to create new schematic components to display several schematic diagrams, including one for the low-voltage subnetwork related to a given feeder.
- In the sixth exercise, you will learn about how to specify parameters to manage association between features and schematic elements so the Propagate Schematic Selection To Map and Propagate Map Selection To Schematic commands, and the Identify tool work in ArcMap.
- During the seventh exercise, you will learn how to specify parameters to relate schematic elements contained in a schematic diagram each other.
- The last exercise concerns application behaviors. The purpose will be to create behaviors that impact schematic diagrams and schematic elements.

Exercise 1: Creating a schematic dataset and editing the schematic project

In this exercise, you will learn how to create the schematic dataset that will contain all the schematic data related to the schematic project you will build. Then, you will learn how to edit the related schematic project parameters within Schematics Designer.

Creating a schematic dataset

A schematic dataset resides in a personal, file or ArcSDE geodatabase; it's the schematic equivalent of a geodatabase feature dataset. The schematic dataset acts as the entry point for your schematic application. A schematic dataset contains a collection of schematic diagram classes and schematic element classes that share the same application domain—for example, water, electrical, and so on.

ArcCatalog is the application used to create schematic datasets. To create a schematic dataset:

1. Start ArcCatalog.
2. Browse to the Schematics_Designer_Tutorial personal geodatabase stored in the C:\arcgis\ArcTutor\Schematics\Designer folder.
3. Right-click this personal geodatabase, point to New, and click Schematic Dataset.

   A new schematic dataset appears in the Catalog tree.
4. Type a new name for the schematic dataset (TutorialII, for example).
5. Press Enter.

Creating a schematic dataset in ArcCatalog automatically creates the associated schematic project where the components that specify the graphical parameters of your schematic application will be stored. These components can be edited and customized using Schematics Designer.

Editing the schematic project within Designer

A schematic project is a set of configurations that provides access to your application's graphical components, including data sources, diagram types, element types, and behaviors.

To define the schematic diagram types and schematic element types related to a schematic dataset, you need to edit the schematic project associated with that schematic dataset within Schematics Designer:

1. Right-click the new schematic dataset in the Catalog tree and click Edit Project.

   The Schematics Designer application starts, loads the associated schematic project parameters in memory, and opens directly in Design mode. As no diagram type already exists in this new schematic dataset, the Diagram Type Assistant automatically launches.
2. Cancel the Diagram Type Assistant for the moment and look at the Designer interface.

When you are working in Design mode, all schematic project components already defined are displayed in the Designer Editor window. The table of contents (tree) on the left side of the window shows all the schematic components that have already been defined for the opened schematic project. The right side of the Designer Editor window shows the current state and values for all parameters that have been defined for the currently selected component.

Working in Design mode allows you to create and modify any project component's parameters (diagram type, element type, and so forth). When components are created, a new entry corresponding to the newly created component is automatically created in the Designer tree.

To enable the Run mode when you are running in Design mode, click the Close button on the bottom right corner of the Designer Editor window. Working in Run mode allows you to test and visualize instantly the schematic project parameters you have just modified when running in Design mode: any parameter change is automatically taken into account and can be graphically visualized. To enable the Design mode when you are running in Run mode, click the Edit tool.

Exercise 2: Setting the schematic project's default parameters

The default parameters for the graphical display of the schematic elements are defined at the Schematic Project level. These include default node symbol, default flag symbol, default text size and text color, default selection style appearance, and other items.

When the Schematic Project entry is selected in the Designer tree, the right side of the window lists all the default project parameters. These parameters can be modified according to your needs. They are regrouped into three tabs:

- The General tab displays where all general graphical project parameters are stored.
- The Effects tab includes line, fill, and text effects used for graphic representations.
- The Highlighting tab regroups parameters related to the display style and mode for elements when they are selected in the diagrams.

In this exercise, default values will be set to:

- Represent the symbol of flags when no specific symbols are defined for their related flag model.
- Represent the symbol of nodes when no specific symbols are defined to represent them.
- Define the symbol scaling factor.
- Define the line effects (color, width) that will be used by default to represent the links.
- Define the text color and the text font that will be used by default to display a label.
- Define the selection style.

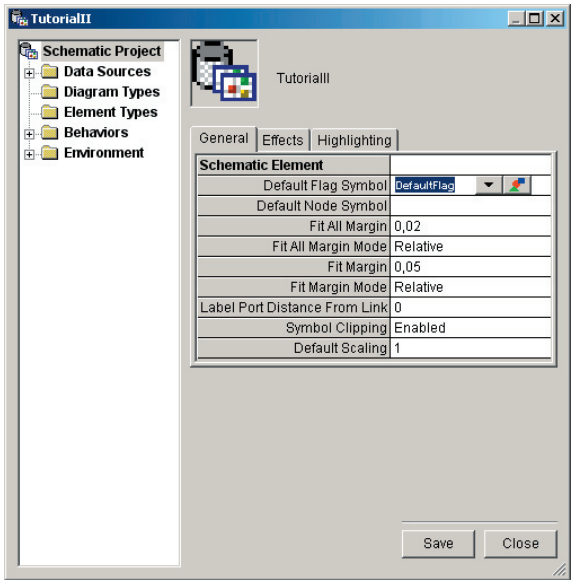
Defining the default flag symbol

The Default Flag Symbol parameter available from the General tab will be the parameter used by default to represent a flag's symbol when this parameter is defined.

1. Click anywhere on the Default Flag Symbol field.
2. Click the NgUSymbolEditor button  that displays to launch the CGM Symbol Editor and Vector Drawing tool.
3. Use the NgUSymbolEditor drawing tools to build your CGM symbol as desired.
4. Click File and point to Save As. Type the name that will reference your first symbol (DefaultFlag, in this example).
5. Exit the NgUSymbolEditor (File > Exit menu).

The Default Flag Symbol field is automatically completed by the name of the symbol you have created.

Note: If you want more details about the NgUSymbolEditor drawing tools, please see the 'NgUSymbolEditor tool' topic in the Schematics Designer book of the ArcGIS Help System.



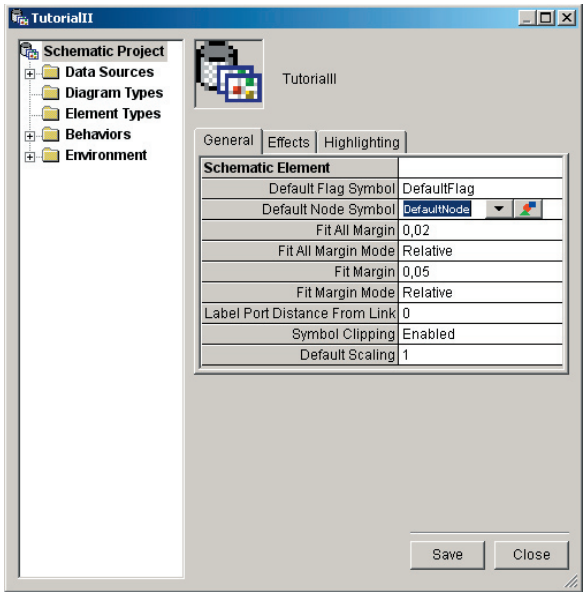
Defining the default node symbol

The Default Node Symbol parameter available from the General tab is the parameter that is used by default to represent schematic nodes when their own Symbol parameter (defined for their related node element type) is not set.

To define this default symbol, follow the steps below:

1. Click the Default Node Symbol field.
2. Click the NgUSymbolEditor button to launch the CGM Symbol Editor and Vector Drawing tool (NgUSymbolEditor).
3. Use the NgUSymbolEditor drawing tools to build your CGM symbol as desired and type a name for the current symbol (File > Save As menu).
4. Close NgUSymbolEditor (File > Exit).

The Default Node Symbol field is automatically completed by the name of the symbol you have just created.

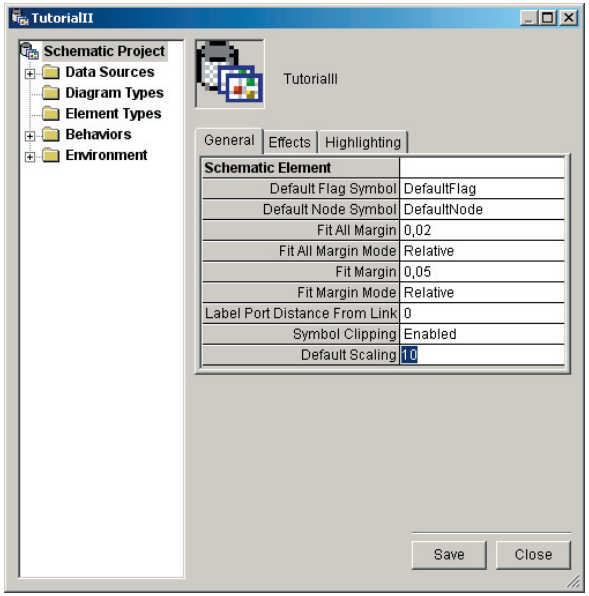


Defining the default scaling

The Default Scaling parameter available from the General tab is the default scale factor that will be applied to the schematic diagram symbols or labels when their own scale parameters are not set.

1. Click the Default Scaling field.
2. Type a value.

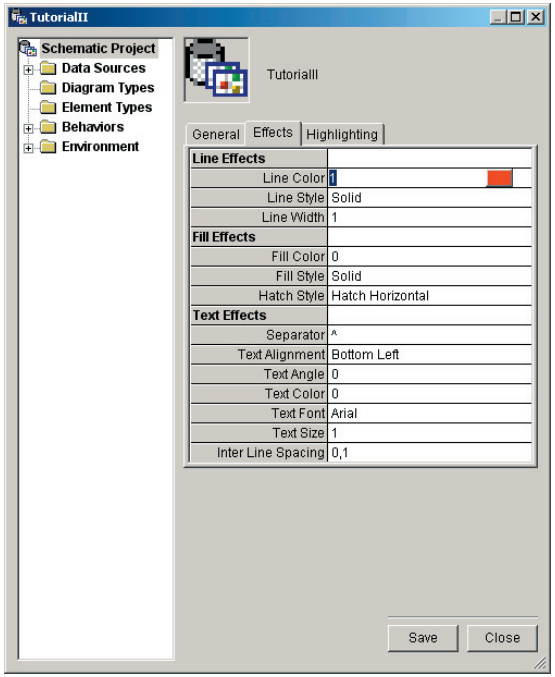
Note: The correct value for your schematic project Default Scaling factor is not easy to determine, and a value that is correct for one schematic project could be inappropriate for another schematic project. In fact, this value depends on your network coordinate system and how close to or far from each other your nodes are in your schematic view. Generally, you will have to test several values before finding the best one.



Defining the default line effects

Parameters defining line effects that will be used by default to represent a link line when these effects are not defined at the link element type level are available from the Schematic Project Effects tab. They are assembled into the Line Effects subset of parameters.

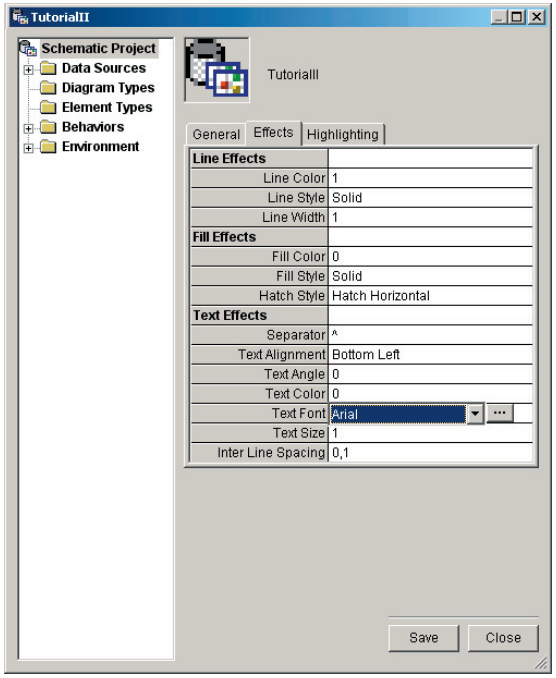
1. Click the Effects tab.
2. Click the Line Color field that will define the default color for the lines, and click the color box that displays to launch the Table Color Editor.
3. Choose or define a color.
4. Click OK to validate.
5. Click the Line Width field to specify the default width for the lines. Modify the value with the one you want.
6. Click the desired default style for lines from the Line Style drop-down list.



Defining the default text effects

Parameters defining text effects that will be used by default to display the labels when these effects are not defined during the label definition are available from the Schematic Project Effects tab. They are assembled into the Text Effects subset of parameters.

1. Click the Effects tab.
2. Click the Text Color field that will define the default color for the labels, and click the color box that displays to launch the Table Color Editor.
3. Choose or define a color and validate.
4. Click the Text Font field that will define the default text font for the labels.
5. Click one of the default fonts already defined from the drop-down list, or click the Ellipsis button displayed on the right to open the Font Editor. The Font Editor lists all fonts installed on your computer and allows you to choose the font you want.



Defining the selection style

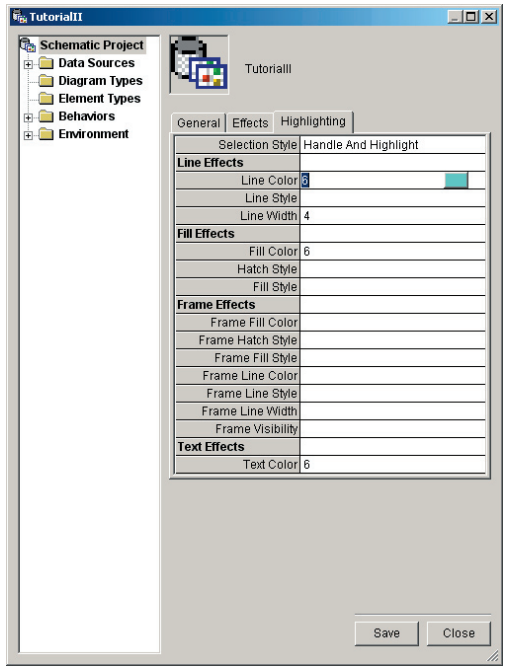
Parameters defining the Selection Style are assembled on the Highlighting tab.

Three selection styles are available:

- With the Handle style, a selected schematic element is singled out by the display of its bounding box.
- With the Highlight style, a selected element is singled out by specific color, width, style, and fill.
- With the Handle And Highlight style, the two Handle and Highlight selection styles are combined.

When the Highlight or Handle And Highlight style is used, some line, fill, and text effects must be set to complete the selection style definition and the appearance of links, nodes, and labels when they are selected in a schematic view.

1. Choose Handle And Highlight from the Selection Style drop-down list.
2. To specify the color for the selected lines, click the Line Color parameter, click the color box to launch the Table Color Editor, and choose or define the desired color that will be used to highlight lines when they are selected.
3. To define the selected lines' width, click the Line Width parameter and type the desired value.
4. For defining the effects for the selected fills, click Solid from the Fill Style drop-down list and define the Fill Color parameter. These fill attributes will be used to display the symbol fill of the nodes that will be selected.
5. To define the effects for the selected labels, click the color box that displays when clicking the Text Color field. In the Table Color Editor that opens, choose or define the color you want.
6. Click OK to validate.



7. Click Save.

   By default, the schematic project definition is automatically checked. This causes the Please verify dialog box to open before the saving is done. This dialog box sums up some points that may have been forgotten during your schematic project building. Note that the Please verify dialog box lists warnings with a critical level mark (red, orange, or green traffic light) without preventing the saving; that is, each warning is only informative and your project definition can be completely operational even if red warnings display in the dialog box.
8. Check Do not perform this verification when saving any more and click Yes, so this verification phase is disabled for the next saving.
9. Click OK to close the dialog box that informs you about how to reactivate the verification phase.

Note: To reactivate the verification phase, you only need to right-click the Schematic Project entry in the Designer tree and check Verify On Save.

Exercise 3: Displaying all network data in a unique schematic diagram

The purpose of this exercise is to create the schematic components required to display all network data stored in the Schematics_Designer_Tutorial database in a unique diagram. The diagram type that will implement this schematic diagram will work with the Custom Query Based Builder. With this predefined builder, you need to specify the queries that return the elements you want to be displayed in your diagrams.

Four main steps are required to create such diagrams:

1. Create a data source required to connect the network data.
2. Create the schematic diagram type that will implement the desired schematic diagram.
3. Create the node element types that will implement the node elements contained in the schematic diagram.
4. Create the link element types that will implement the link elements contained in the schematic diagram.

Creating the data source

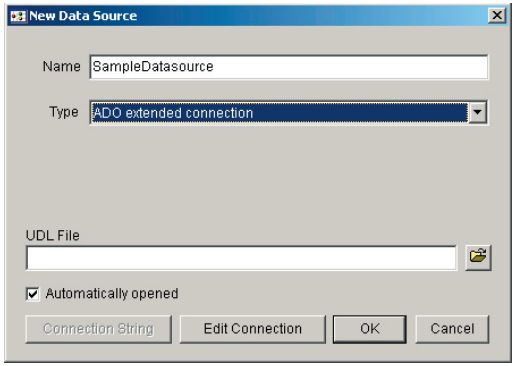
A data source is related to a database where network elements that you want to be displayed in your schematic diagrams are stored.

By default, when you create the schematic dataset in the Schematics_Designer_Tutorial database, Schematics automatically created a data source named CURRENTDS. The CURRENTDS data source is an ESRI GDB Datasource that connects the geodatabase, which contains the schematic dataset related to the schematic project currently open in Designer. This data source is always available in the Designer tree; it displays under the Data Sources entry. You can use the CURRENTDS data source when you create new element types or diagram types related to data stored in your geodatabase. But if you need to connect your geodatabase using a specific provider or if you need to connect another database (for example, a database that contains inside plants' data), you will have to create a new data source.

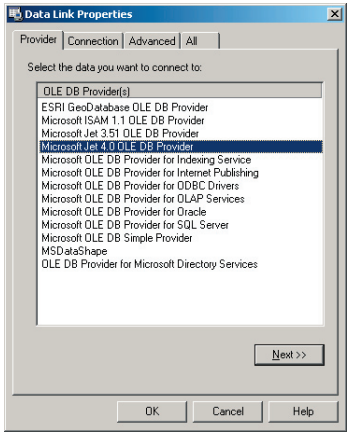
The ESRI data source types are not always easy to use for SQL queries that join several tables. In this tutorial, queries will be correctly interpreted using the CURRENTDS data source, but as an example, the following steps show how to create an ADO extended connection data source using the Microsoft Jet 4.0 OLE DB provider which can be used to work with any Access GDB.

Note: The following 9 steps are just sample steps to learn about how to create a datasource within ArcGIS Schematics Designer. These steps are not needed for the rest of the tutorial.

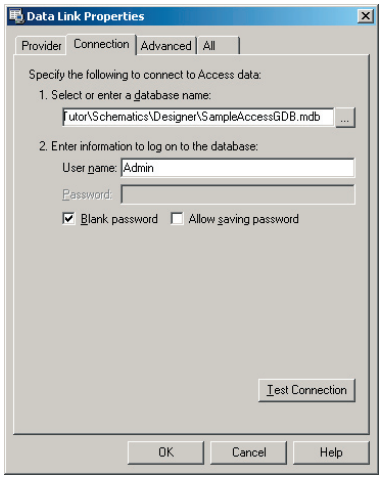
1. Right-click Data Sources in the Designer tree and point to Add Data Source.
   The New Data Source dialog box opens.
2. Type a name for your new data source in the Name field.
3. Click ADO extended connection from the Type drop-down list.



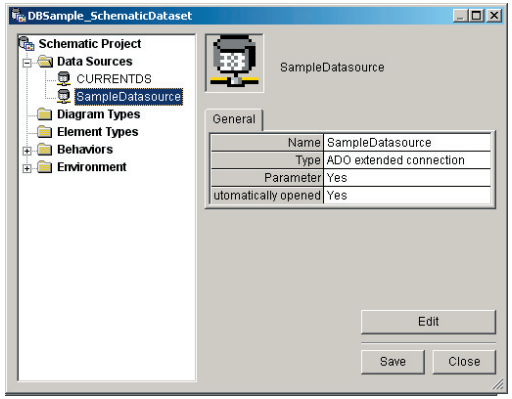
4. Click Edit Connection to specify the connection parameters.
   The Microsoft Data Link Properties dialog box opens.
5. Click the Provider tab and click Microsoft Jet 4.0 OLE DB Provider from the list.



6. Click the Connection tab and click the Ellipsis button displayed on the right side of the Select or enter a database name area.
   The Select Access Database dialog box opens.
7. Browse to your Access database and click Open. The Select Access Database dialog box closes.
   The Data Link Properties dialog box should appear as follows:



8. Click Test Connection if you want to check that the connection parameters are correctly specified. Click OK to close the Data Link Properties dialog box.
9. Click OK to close the New Data Source dialog box. The newly created data source should display in the Designer tree as follows:



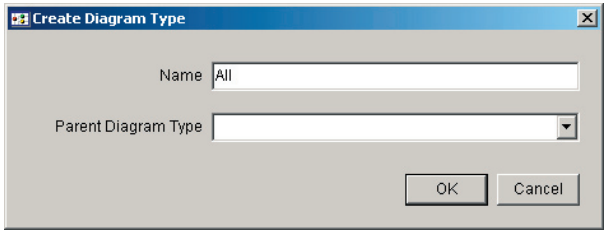
Creating the first schematic diagram type

A network is made up of nodes and links. These nodes and links are displayed in different generic structures known as diagrams. A diagram is a collection of schematic views and is related to a single schematic project. A diagram type is a template that brings together a set of diagrams that have common characteristics. More than one diagram type can be created for a schematic project. Typically, one diagram type can be defined for a specific view of the network. For example, a diagram type is defined for a schematic of the electric network of primaries and switches, and a second diagram type is defined for a schematic of the internal schema (inside plant) of the devices. You can either create a diagram type that inherits another diagram type or create a diagram type that inherits no parent diagram type.

The purpose of this step is to define the schematic diagram type that will implement a unique schematic diagram that is, the schematic diagram where all network data stored in the Schematics_Designer_Tutorial tutorial database will be displayed.

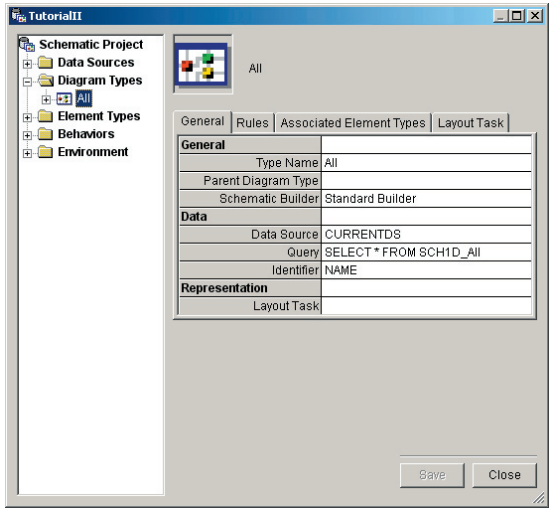
1. Right-click Diagram Types in the Designer tree and click Create.
   The Create Diagram Type dialog box opens.
2. Type the name that will be used to reference the new diagram type (All for example).

Since no diagram has already been created, this new diagram type cannot inherit from another one (no Parent Diagram Type can be set).



3. Click OK to validate.

The new diagram type subentry is automatically created below the Diagram Types main entry in the Designer tree.

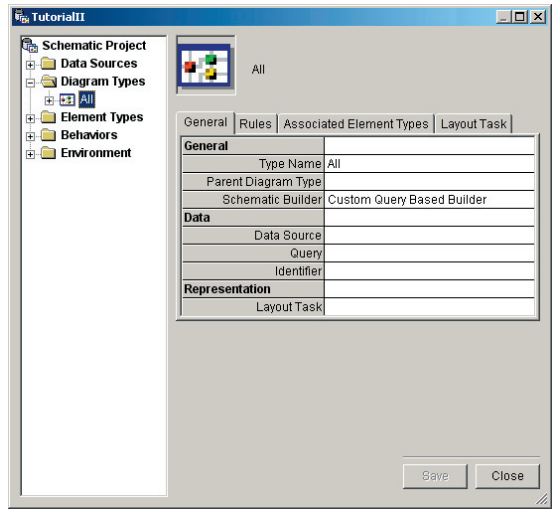


When a new diagram type is created, Schematics specifies the Standard Builder by default for the Schematic Builder field; it automatically creates the two DIAGRAMCLASSID and DIAGRAMOBJECTID attributes and sets some parameters for the newly created diagram type.

In this exercise, you have to create a schematic diagram type that will work with the Custom Query Based Builder. Moreover, because this diagram type will implement a unique schematic diagram, no query, identifier, or data source is required.

4. Change Standard Builder to Custom Query-Based Builder from the Schematic Builder drop-down list.
5. Click the Query field and click the Ellipsis button that displays. The Query Editor dialog box opens.
6. Select the SELECT * FROM SCH1D_All query specified in the Query area and remove it.
7. Click OK to close the Query Editor dialog box.
   The Query field on your diagram type's General tab should now be empty.
8. Click the Identifier field and click the Ellipsis button that displays to open the Identifier Editor dialog box.
9. Click NAME from the Identifier area and click the Up arrow to remove the NAME field from this area.
10. Click OK to close the Identifier Editor dialog box.
    The Identifier field on your diagram type's General tab should now be empty.
11. Specify no data source for the Data Source field.
12. Expand the Attributes entry under the Diagram Types entry, right-click DIAGRAMCLASSID, and click Delete. This predefined attribute is removed.
13. Proceed in the same way to remove the DIAGRAMOBJECTID second predefined attribute.

At the end of these steps, your Diagram Types entry should display as follows:



Creating the node element types

An ArcGIS junction is implemented as a Schematics node element. The node represents a device or other point feature of a network. It is displayed as a symbol and is assigned graphic (color, fill style, and so forth), geometric (scaling factor, rotation, and so forth), and visual (visibility, highlighted state, and so forth) parameters. Any node is implemented by a schematic node element type that holds the common property of a set of nodes.

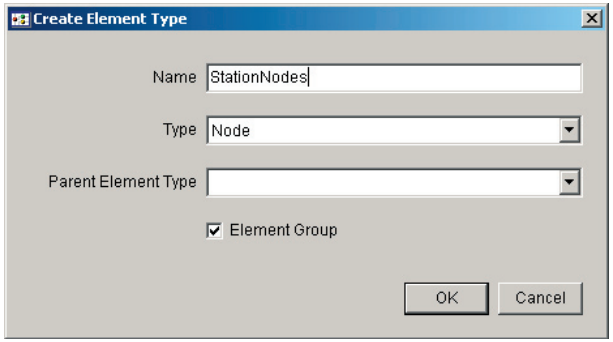
The purpose of the following steps is to create two schematic node element types: one to implement the display of all records stored in the Station table and another to implement the Feeder table's records.

Creating the StationNodes element type

1. Right-click Element Types in the Designer tree and click Create. The Create Element Type dialog box opens.
2. Type the name that will be used to reference your first schematic node element type (StationNodes in this example).
3. Click Node from the Type drop-down list.

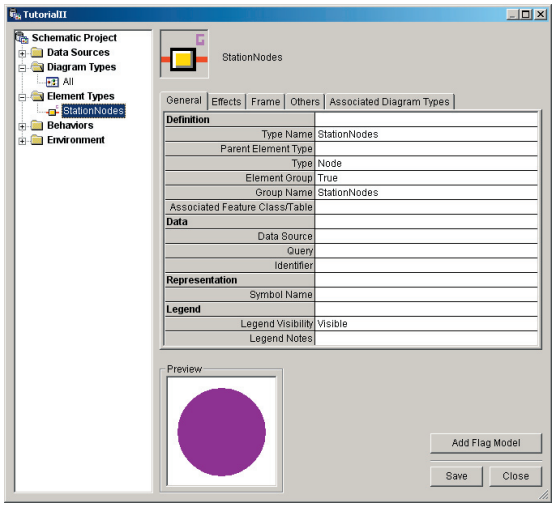
Since no node element type has already been created for this schematic project, this new node element type cannot inherit from another one (no Parent Element Type can be set).

4. Keep the Element Group option checked by default to ensure you can customize your element type display afterward.



5. Click OK to validate.

The new node element type subentry is automatically created below the Element Types main entry in the Designer tree.



The Preview window shows the symbol that will be used by default to represent all nodes of this type. In this case, it is the schematic project's default node symbol you defined in exercise 2.

The purpose now is to set the parameters for this new node element type and, in particular, set parameters to query and identify each node this StationNodes element type will implement. These parameters are available from the General tab, and they are assembled into the Data subset of parameters.

6. Click the General tab and click the CURRENTDS data source from the Data Source drop-down list.
7. Click the Query field and click the Ellipsis button that displays. The Query Editor dialog box opens.
8. From the Tables list, click the Station table where all records corresponding to the network stations are stored.
9. Click All Records.
   The SELECT * FROM Station SQL query automatically displays in the Query area.
10. Click OK to validate.

The Query Editor dialog box closes. The specified query automatically displays on the General tab's Query parameter field.

11. Click the General tab's Identifier field and click the Ellipsis button that displays.
    The Identifier Editor dialog box opens.

The Fields list on the Identifier Editor dialog box displays all fields returned by the query you have just specified. These fields can be used to identify the schematic elements of the element type.

The SCH_DiagramType and SCH_DiagramName items are also available at the end of this list. These predefined items are related to the diagram type name and the diagram name where schematic elements will be displayed. They can help you ensure that each element identifier contained in your schematic diagram is unique.

In this example, you will identify each StationNodes node by concatenating the diagram type name, the diagram name, the Station string, and the OBJECTID field values.

Note: Using the Station string in the node identifier building is useful in this example because the FromJunctionType and ToJunctionType fields available from the tables of links specify whether each link connects a Station type node or a Feeder type node. By using the Station string in your StationNodes nodes' identifiers, it will be easy to specify the origin and extremity node for your links afterward.

12. Double-click the SCH_DiagramType and SCH_DiagramName fields available from the Fields list so these two items appear in the Identifier list.
13. Type "Station" in the central text zone and validate this string by clicking the Down arrow.
14. Double-click the OBJECTID field from the Fields list to select it and click OK.

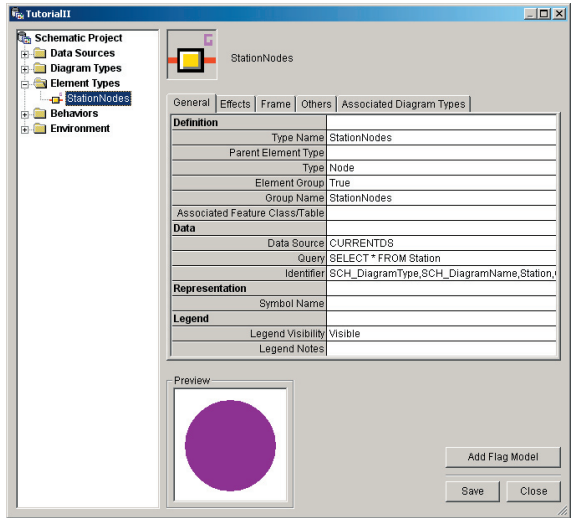
The specified items automatically display in the General tab's Identifier parameter field.

When specifying Identifier for an element type, don't forget these two rules:

- All the elements related to an element type must be identifiable as a unique element.
- Two elements of the same type (node, link, and so forth) cannot have the same identifier when they are found in the same diagram.

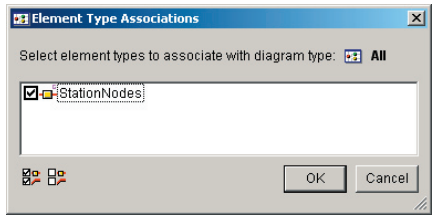
Note: For more information about the Query Editor and Identifier Editor, open the Schematics Designer book in the ArcGIS Help System, load the Setting the node element type General parameters topic, and click the Query Editor and Identifier Editor hyperlinks in the Data section.

At the end of these steps, your first node element type entry should display as follows:



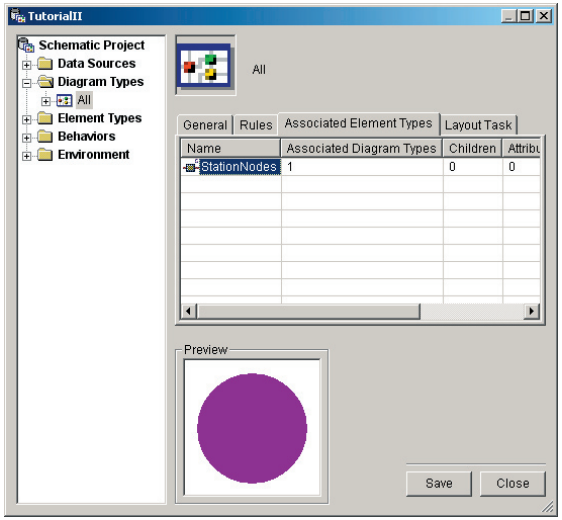
Previously, you created the All schematic diagram type that will implement one schematic diagram where all network data stored in the Schematics_Designer_Tutorial database will be displayed. The purpose now is to specify that the nodes implemented by the StationNodes node element type you have just defined must be displayed in this schematic diagram. To specify the schematic elements you want to be contained in a schematic diagram, you must associate the desired element types with this diagram type:

15. Right-click the All diagram type entry in the Designer tree and click Edit Associations.
    The Element Type Associations dialog box automatically opens.



16. Check the StationNodes item.
17. Click OK to validate.

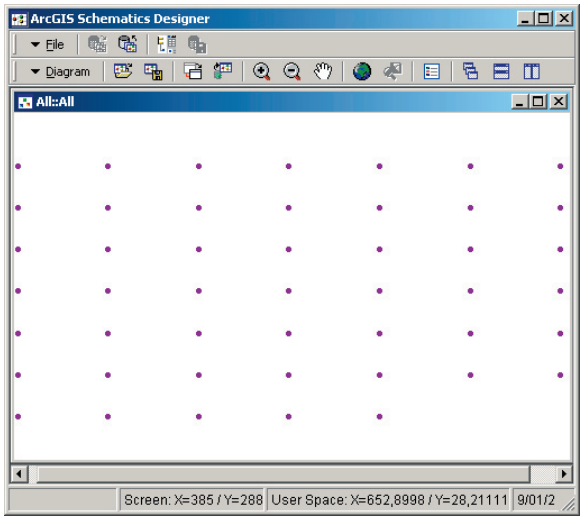
The new Associated Element Types tab should appear for your diagram type. This tab lists all the schematic element types that are currently associated with a diagram type. At the same time, for your StationNodes element type, the new Associated Diagram Types tab has been created; this tab lists all the diagram types with which that element type is associated.



It is time now to test that your StationNodes node correctly displays in your schematic diagram.

18. Click Save displayed at the bottom right corner of the Designer Editor window to save all the defined parameters.
19. Click Close to close the Designer Editor window as well as to stop the Design mode. You are now in Run mode.
20. Click File and click Diagram Toolbar.
    The Diagram toolbar displays.
21. Click the Open Diagrams button  .
    The Select Diagram To Open dialog box opens. The only diagram type, All, is available from the Diagram Type drop-down list, and the All diagram is available from the Diagram Name drop-down list.
22. Click OK to validate.

Your first schematic diagram opens. Since no geometric attributes are defined for any node (that is, x,y coordinates are not specified for each node), Designer automatically places the nodes on a default grid.



Creating the FeederNodes element type

The purpose now is to create the second node element type that will implement the nodes related to the records stored in the Feeder table. Before creating this new schematic component, you must stop the Run mode and switch to Design mode:

1. Close the open diagram and click the Edit button to reedit your schematic project parameters.

Then, create your new node element type as follows:

2. Right-click Element Types in the Designer tree and click Create.
3. In the Create Element Type dialog box that opens, type a name for the new node element type (FeederNodes for example).
4. Click Node from the Type drop-down list.
5. Ensure that the Element Group check box is checked and click OK.

Under the Element Types entry, the new FeederNodes node element type entry should have been automatically created.

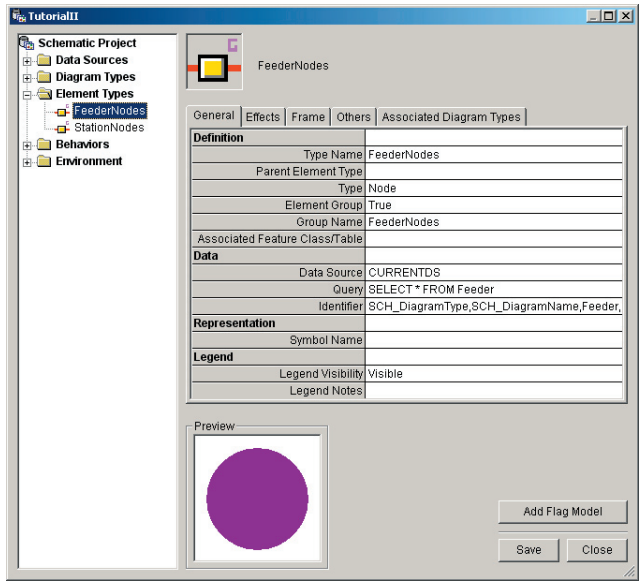
Now, complete your FeederNodes node element type definition by specifying the data source and the query that will return all feeder nodes stored in the Schematics_Designer_Tutorial database and by identifying all nodes the FeederNodes node element type will implement.

6. Click FeederNodes in the Designer tree.
7. Click CURRENTDS from the Data Source drop-down list.
8. Click the Query parameter field and click the Ellipsis button that displays to open the Query Editor dialog box.
9. Double-click the Feeder table from the Tables list so the SELECT * FROM Feeder SQL query automatically appears in the Query field.
10. Click OK.
    The Query Editor dialog box closes.
11. Click the Identifier parameter field and click the Ellipsis button that displays to open the Identifer Editor dialog box.

As you had previously done for the StationNodes element type identifier, each FeederNodes node will be identified by concatenating the diagram type name and the diagram name where it will be displayed, the Feeder string, and the OBJECTID field values.

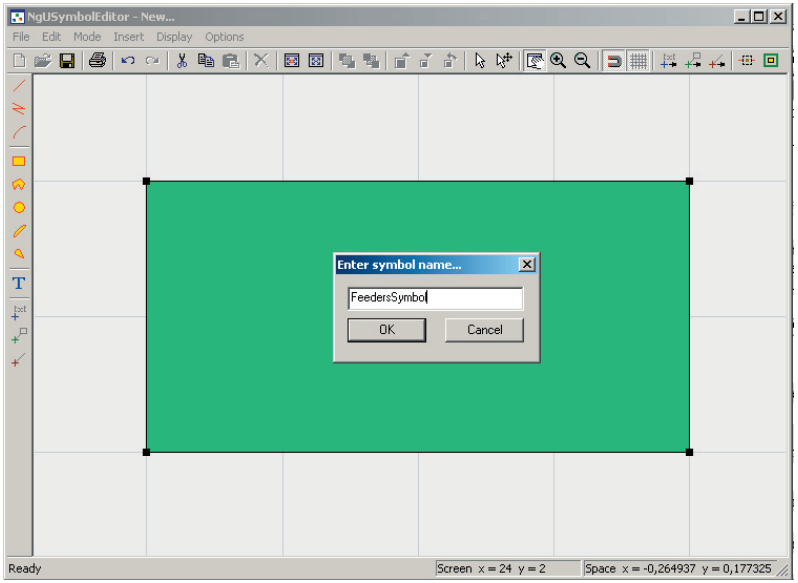
12. Double-click the SCH_DiagramType and SCH_DiagramName items available at the end of the Fields list so these two fields appear in the Identifier list.
13. Type “Feeder” in the central text zone and validate by clicking the Down arrow.
14. Double-click the OBJECTID field from the Fields list to add this field to the Identifier list.
15. Click OK.
    The Identifier Editor dialog box closes.

At the end of these steps, the General tab related to your new FeederNodes element type should display as follows:



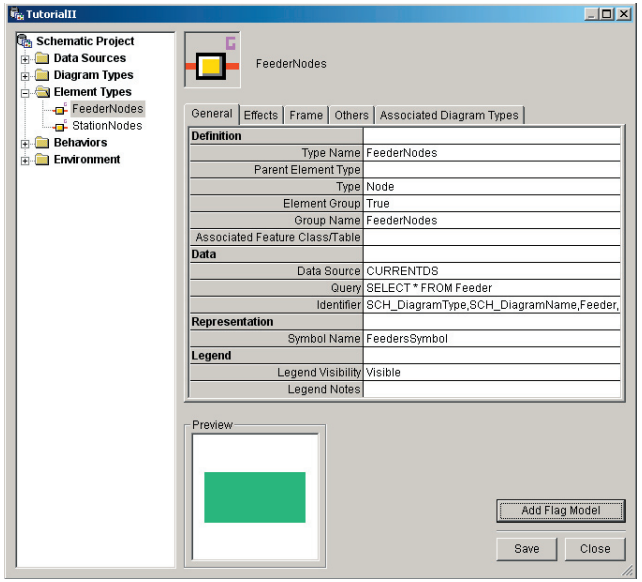
For the StationNodes element type, for instance, no specific CGM symbol has been defined to represent station nodes, so the default symbol defined at the Schematic Project level is the symbol used by default. For the FeederNodes element type, a specific CGM symbol will be created. This symbol is defined when the Symbol Name parameter available from the General tab is set.

16. Click the Symbol Name parameter field and click the NgUSymbolEditor button that displays to launch the CGM Symbol Editor and Vector Drawing tool.
17. Draw your symbol.
18. Click File and click Save.
19. Type a name for the newly created symbol.



20. Click File and click Exit.
    The NgUSymbolEditor drawing tools toolbox closes.

The Symbol Name field on the FeederNodes General tab is automatically completed by the name of the symbol you have created. The Preview subwindow on the left corner of the Designer Editor window shows this new symbol. The General tab related to your Feeders node type should appear as follows:



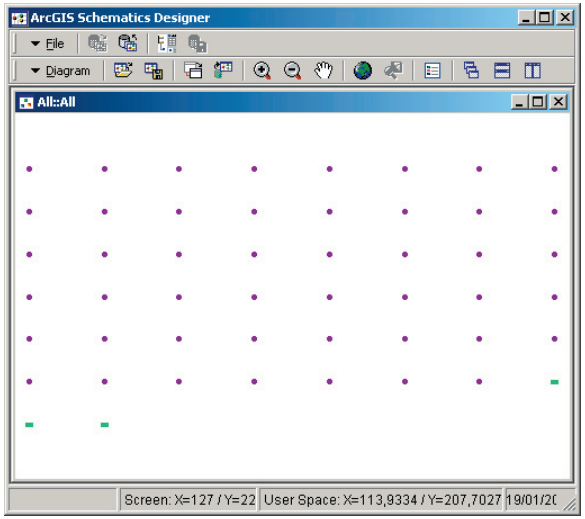
You must now associate the newly created FeederNodes element type with the All diagram type so that all nodes (station and feeder) stored in the Schematics_Designer_Tutorial database display in the diagram that the All diagram type implements:

21. Right-click the All diagram type entry and click Edit Associations.
22. Check the FeederNodes item in the list and validate.

The FeederNodes item automatically adds onto the Associated Element Types tab related to the All diagram type.

23. Click Save to save the newly defined parameters.

If you want to test that the nodes implemented by the new FeederNodes element type correctly display in your schematic diagram, close the Designer Editor window and click the Open Diagrams button. Click the All diagram from the Select Diagram To Open dialog box that displays and validate. The All schematic diagram should appear as follows:



Close the open diagram and click the Edit button to switch to Design mode.

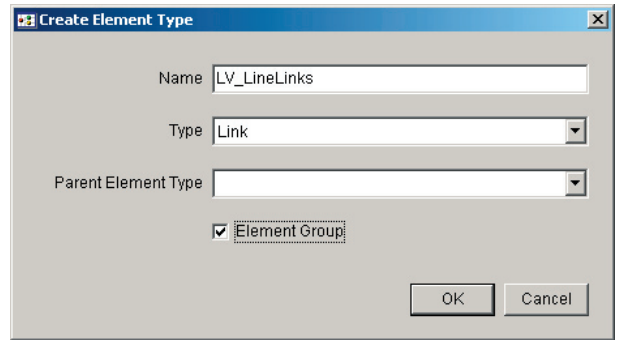
Creating the link element types

An ArcGIS edge is implemented as a Schematics link element. The link represents an edge of a network. A link is represented either by a direct single line segment or by several line segments passing through one or more link points known as vertices. Line patterns and line styles (color, style, width) can be applied to links to refine their graphical display. Any link is implemented by a schematic link element type that holds the common property of a set of links.

The purpose of the following steps is to create two schematic link element types: one to implement the display of all records stored in the LV_Line table and another to implement the HV_Line table's records.

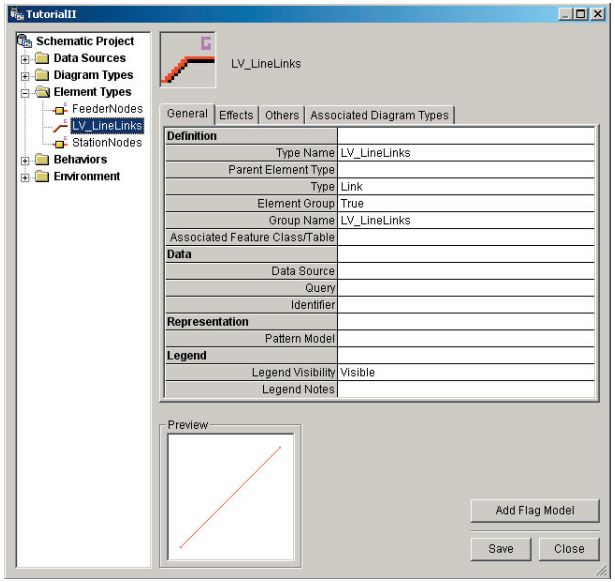
Creating the LV_LineLinks element type

1. Right-click Element Types in the Designer tree and click Create. The Create Element Type dialog box opens.
2. Type a name for the link element type (LV_LineLinks for example).
3. Click Link from the Type drop-down list.



4. Ensure the Element Group check box is checked and click OK.

Under the Element Types entry, the new LV_LineLinks element type entry should have been automatically created. The Preview window shows the link appearance for the links of this type. Since no specific values are specified for the line display, the schematic project's default parameter values set for lines are used.



The purpose now is to set the parameters for this new link element type and, in particular, set parameters to query and identify each link this LV_LineLinks element type will implement. These parameters are available from the General tab, and they are assembled into the Data subset of parameters.

5. Click LV_LineLinks under the Element Types main entry.
6. Click CURRENTDS from the Data Source drop-down list.
7. Click the Query parameter field and click the Ellipsis button that displays to open the Query Editor dialog box.
8. From the Tables list, click the LV_Line table where all records corresponding to the network LV_Line links are stored.
9. Click All Records.
   The SELECT * FROM LV_Line SQL query automatically displays in the Query area.
10. Click OK to validate and close the Query Editor dialog box.

The specified query automatically displays in the General tab's Query parameter field.

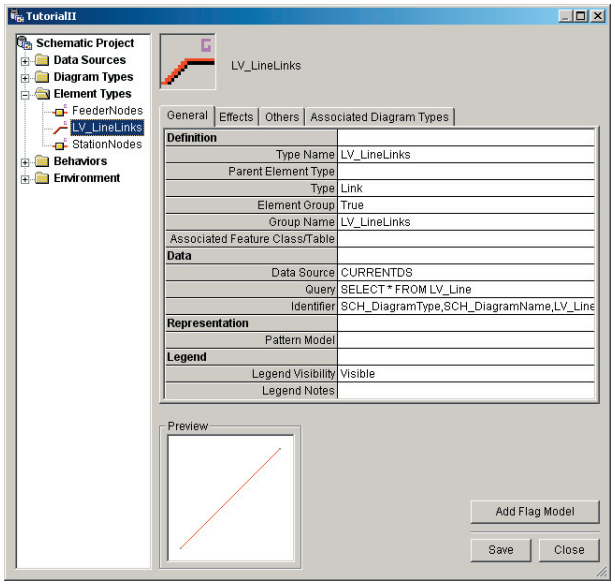
11. Click the Identifier field and click the Ellipsis button that displays. The Identifier Editor dialog box opens.

To be homogeneous with the node elements identifying process, each LV_LineLinks element identifier will be built according to the same format: The diagram type name, the diagram name, the LV_Line string value, and the OBJECTID field values will be concatenated to build the identifiers.

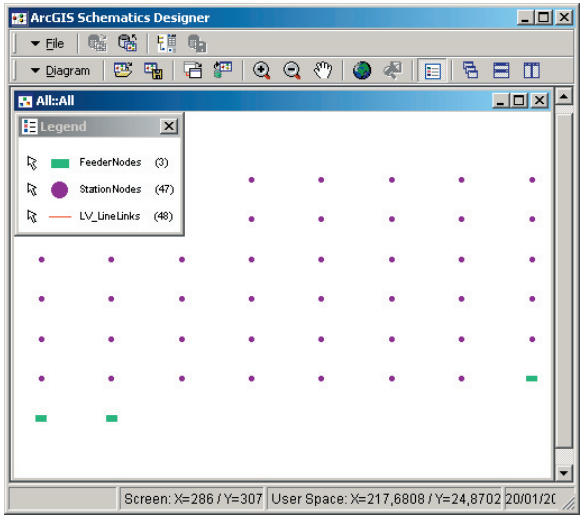
12. Double-click the SCH_DiagramType and SCH_DiagramName items available at the end of the Fields list so these two items appear in the Identifier list.
13. Type "LV_Line" in the central text zone and validate this string by clicking the Down arrow.
14. Double-click the OBJECTID field from the Fields list to select it and click OK.

The Identifier Editor dialog box closes and the specified items automatically display in the General tab's Identifier field.

At the end of these steps, the General tab related to the LV_LineLinks element type should display as follows:



Suppose that you want to test the display of your network. Even if you associate your new link element type with your schematic diagram type (this step is essential to display your new links), your network will appear as follows:

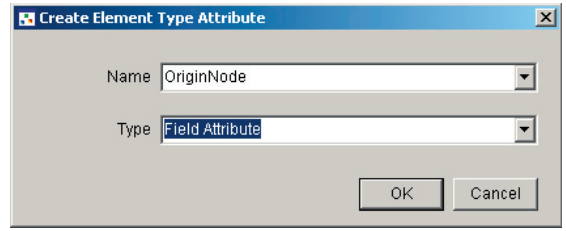


Note that even though the links appear in the Legend subwindow, they are not displayed. The schematic links' display is conditioned by their topological properties; that is, origin and extremity schematic nodes must be specified to display a schematic link. To specify this information, you will have to create two specific attributes for your link element type. These attributes must be created with the OriginNode and ExtremityNode predefined names. When such attributes are created for a link element type, their values should return the origin node identifier and the extremity node identifier that connect each link of this element type. The purpose of the following steps is to define these two compulsory attributes for the LV_LineLinks element type:

15. Right-click LV_LineLinks in the Designer tree and click Create Attribute.
    The Create Element Type Attribute dialog box opens.
16. Click OriginNode from the Name drop-down list.

Then, specify the type of the attribute. In this case, because origin node identifiers can be retrieved from the fields stored in the LV_Line table and since these fields are returned by the query defined for the LV_LineLinks element type, you have to create a field attribute.

17. Click Field Attribute from the Type drop-down list.

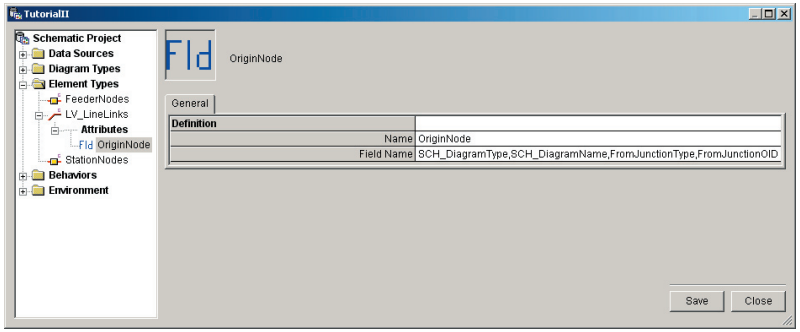


18. Click OK.

The Identifier Editor automatically opens. It allows you to complete the attribute definition by setting the field or the set of fields returned by the LV_LineLinks element type's query that will be merged to compute each origin node identifier. The OriginNode attribute format should follow the node identifier format. In this example, each node identifier merges its diagram type name, its diagram name, a string value corresponding to the type of node (Station or Feeder), and its OBJECTID field value. The OriginNode attribute values must be merged with this information:

19. Double-click the SCH_DiagramType and SCH_DiagramName items at the end of the Fields list so these two items appear in the Identifier list.
20. Double-click FromJunctionType from the Fields list so this specific field takes part in the identifying process (this field, stored in the LV_Line table, identifies the link's origin node type; that is, Station or Feeder node).
21. Repeat this operation with the FromJunctionOID field (this field, stored in the LV_Line table, identifies the link's origin node OBJECTID).
22. Click OK to validate.

At the end of these steps, the OriginNode attribute's General tab should display as follows:



Then, following the same process, you will create the ExtremityNode attribute.

23. Right-click LV_LineLinks and click Create Attribute to open the Create Element Type Attribute dialog box.
24. Click ExtremityNode from the Name drop-down list.
25. Ensure the Field Attribute value is selected in the Type drop-down list and click OK.
26. Then, from the Identifier Editor dialog box that opens, double-click the SCH_DiagramType and SCH_DiagramName items available at the end of the Fields list so these two items appear in the Identifier list.
27. Double-click ToJunctionType from the Fields list so this specific field takes part in the identifying process (this field, stored in the LV_Line table, identifies the link's extremity node type; that is, Station or Feeder node).
28. Repeat this operation with the ToJunctionOID field (this field, stored in the LV_Line table, identifies the link's extremity node OBJECTID).
29. Click OK to validate.

The ExtremityNode attribute definition is now complete.

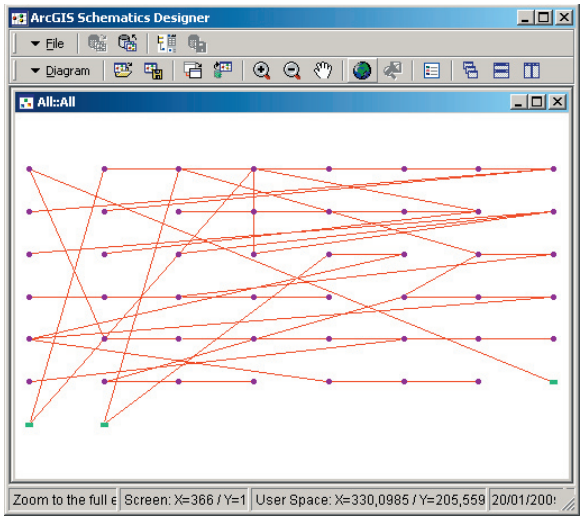
The LV_LineLinks element type definition is now complete. Now you only need to associate this new link element type with the All diagram type so the links implemented by the LV_LineLinks element type are displayed in the diagram implemented by the All diagram type.

As previously done when you associated your element node types with your diagram type:

30. Right-click the All diagram type entry in the Designer tree and click Edit Associations.
31. Click the LV_LineLinks from the Name drop-down list and validate.

The LV_LineLinks item automatically adds onto the All diagram type's Associated Element Types tab.

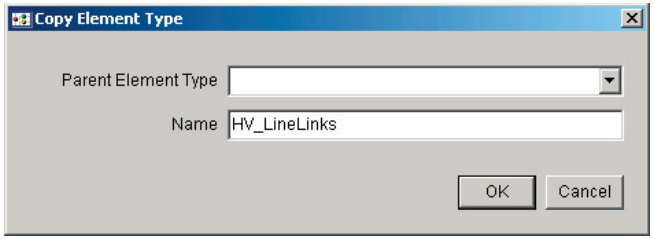
If you switch to Run mode and open the All diagram, it should display as follows:



Creating the HV_LineLinks element type

The purpose now is to create the second link element type that will implement the links related to the records stored in the HV_Line table. Because this new link element type definition will be very close to the LV_LineLinks element type you have just created, there is a quick way to proceed by copying the LV_LineLinks element type definition:

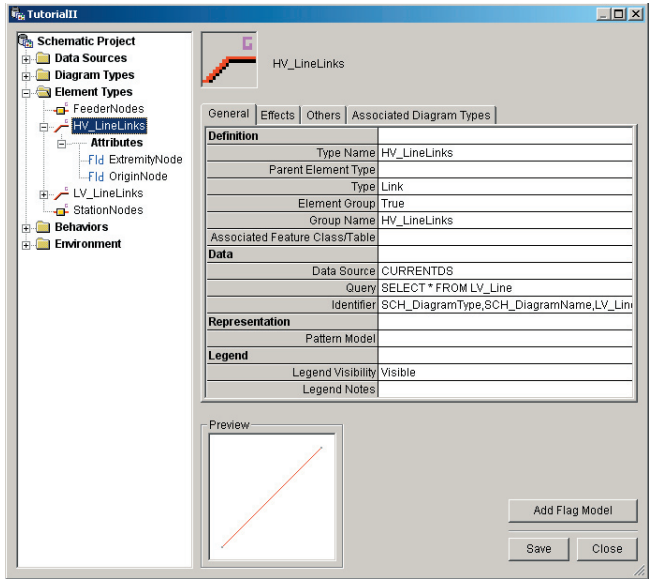
1. If you are in Run mode, start by closing the open diagram and clicking the Edit button to switch to Design mode.
2. Right-click LV_LineLinks in the Designer tree and click Copy. The Copy Element Type dialog box opens.
3. Keep the Parent Element Type field empty.
4. Type a name for your new link element type (HV_LineLinks in this example)



5. Click OK.

The Copy Element Type dialog box closes. The new HV_LineLinks element type entry displays under the Element Types entry in the Designer tree.

All general parameters (query, identifier) and attributes defined for the LV_LineLinks element type from which the new HV_LineLinks element type has been copied are automatically reported. The HV_LineLinks element type entry presents as follows:



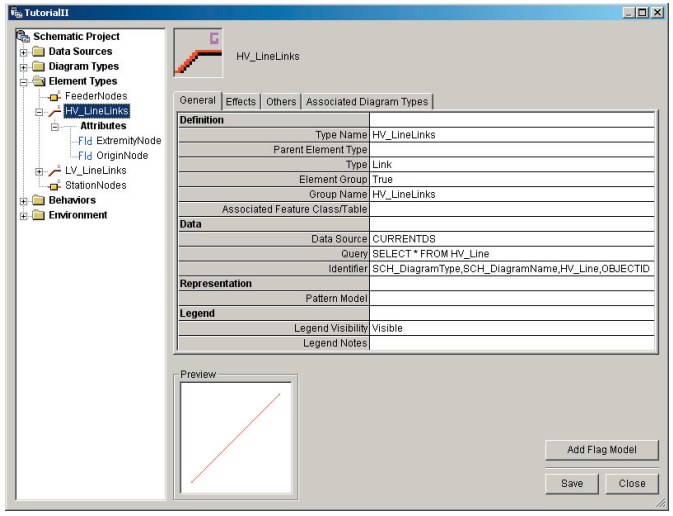
You must now modify the definition of the HV_LineLinks element type so it is related to the HV_Line table instead of the LV_Line table.

6. Click the HV_LineLinks element type's General tab.
7. The copied data source is OK.
8. Then, notice the Query field. The copied query returns all the links stored in the LV_Line table. You must modify this query so it returns all the links stored in the HV_Line table. Click the Query parameter field and click the Ellipsis button that displays to open the Query Editor dialog box.
9. Remove the query from the Query field.
10. Double-click the HV_Line table from the Tables list so the SELECT * FROM HV_Line SQL query automatically appears in the Query area.
11. Click OK to validate.
    The Query Editor dialog box closes. The Query field value on the HV_LineLinks' General tab is updated with the newly specified query.

As you had done for all previously defined element types, each HV_LineLinks link will be identified by concatenating the diagram type name and the diagram name where it will be displayed, the HV_Line string, and the OBJECTID field values. The copied Identifier must be modified:

12. Click the Identifier parameter field and click the Ellipsis button that displays to open the Identifier Editor dialog box.
13. Select the OBJECTID item from the Identifier list and click the Up arrow to remove the item from the list.
14. Proceed in the same way to remove the LV_Line item from the Identifier list.
    Then, keep both SCH_DiagramType and SCH_DiagramName items in the Identifier list and complete the definition as follows:
15. Type “HV_Line” in the central text zone and validate by clicking the Down arrow.
16. Double-click the OBJECTID field from the Fields list to add this field to the Identifier list.
17. Click OK.

The Identifier Editor dialog box closes and the specified items automatically display in the General tab’s Identifier field. At the end of these steps, the General tab related to the HV_LineLinks element type should appear as follows:



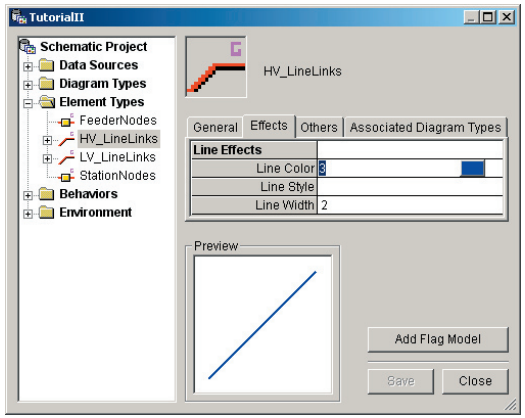
Now, notice the OriginNode and ExtremityNode attributes that have been copied for the HV_LineLinks element type during the copying operation. These OriginNode and ExtremityNode attribute formats must follow the node identifier format that, in this example, merges the diagram type name and the diagram name where these nodes display, the Station or Feeder string, depending on the node type, and the node OBJECTID. All this information is available from the HV_Line table, and all the HV_Line table’s fields are returned by the query defined for the HV_LineLinks element type. You only need to check that the fields used to generate these attributes are correctly specified:

18. From the OriginNode attribute’s General tab, the Field Name field indicates how the attribute values will be built. It is obtained by concatenating SCH_DiagramType, SCH_DiagramName, the FromJunctionType, and the FromJunction OID fields. The information is specified as expected.
19. Then, check the General tab’s Field Name field value for the ExtremityNode attribute. The concatenated items are also specified as expected.

For the LV_LineLinks element type definition, no graphic effects had been defined to customize the link display. In this case, since no specific values are specified for the line color, line width, and line style parameters from the Effects tab, the schematic project’s default parameter values set for lines are used. For the HV_LineLinks element type, it could be interesting to customize its own line parameters so this link’s display is different from the LV_LineLinks links.

20. Click HV_LineLinks in the Designer tree and click the Effects tab.
21. Click the Line Color field and click the color box that displays to launch the Color Editor.
22. Choose the desired color and click OK to validate.
    The Color Editor closes. The Line Color parameter on the Effects tab references the newly specified color.
23. Type a value in the Line Width field if you want to change the width of the lines for the HV_LineLinks element type.

The following screen shot is a sample of the Effects tab's content after these steps:

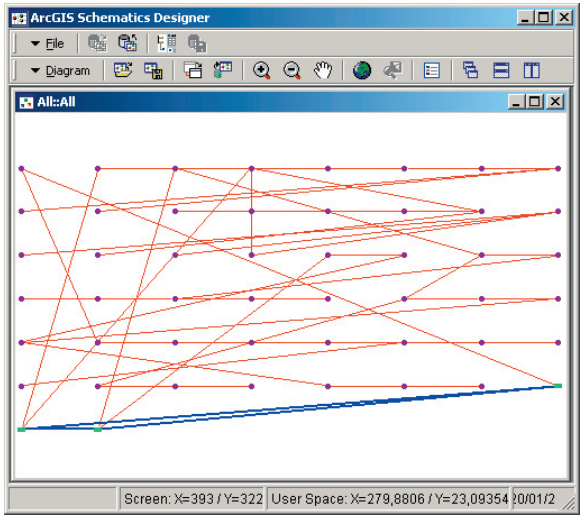


The HV_LineLinks element type definition should be correct.
You must now associate this new element type with the All diagram type before testing that your HV_LineLinks links correctly display in the diagram.

24. Right-click the All diagram type entry in the Designer tree and click Edit Associations.
25. Check the HV_LineLinks item and click OK.

The HV_LineLinks item automatically adds onto the All diagram type's Associated Element Types tab.

If you switch to Run mode and open the All diagram, it should display as follows:



Displaying schematic elements using their geographic coordinates

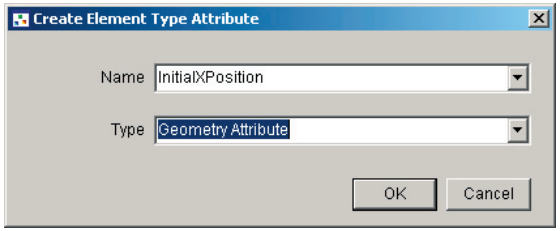
Until now, since no geometric attribute is defined for the node element types (that is, x and y coordinates are not specified for the schematic node elements), ArcGIS Schematics automatically places the nodes contained in the All diagram on a default grid. The purpose now is to create two specific attributes for the StationNodes and FeederNodes element types so the node elements implemented by these element types display at their geographic positions. These two attributes are InitialXPosition and InitialYPosition.

Generally, InitialXPosition and InitialYPosition attributes must be linked to specific fields stored in the database.

- If your nodes' x and y coordinates are clearly stored in two fields in your database, you can define these attributes, such as simple attribute fields.
- If no x and y fields are clearly available in your database tables and if the data source used to return the node elements from this database is an ESRI data source, ArcGIS Schematics can decode the information from the ESRI Shape field using a geometry attribute.

In this example, the StationNodes and FeederNodes element types' specified queries return the node elements from the Station and Feeder tables using the CURRENTDS data source. As this data source is an ESRI GDB data source, you can create geometry attributes for defining the InitialXPosition and InitialYPosition attributes.

1. Right-click the FeederNodes element type entry and click Create Attribute. The Create Element Type Attribute dialog box opens.
2. From the Name drop-down list, click InitialXPosition.
3. From the Type drop-down list, click geometry attribute.

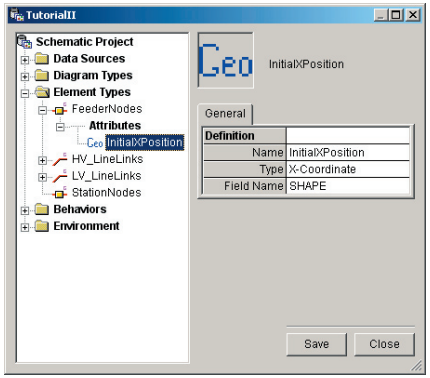


4. Click OK.

The Create Element Type Attribute dialog box closes. A new Attributes entry is automatically created below the FeederNodes element type entry, and the newly created InitialXPosition attribute is referenced below this new entry.

5. From the new attribute's General tab, keep the default value for the Type parameter (X-coordinate).
6. From the Field Name drop-down list, click SHAPE and validate.

At the end of these steps, the InitialXPosition attribute's General tab content should appear as follows:



In a similar way, you need to repeat the process to create the InitialYPosition attribute required to display the FeederNodes according to their y-coordinates.

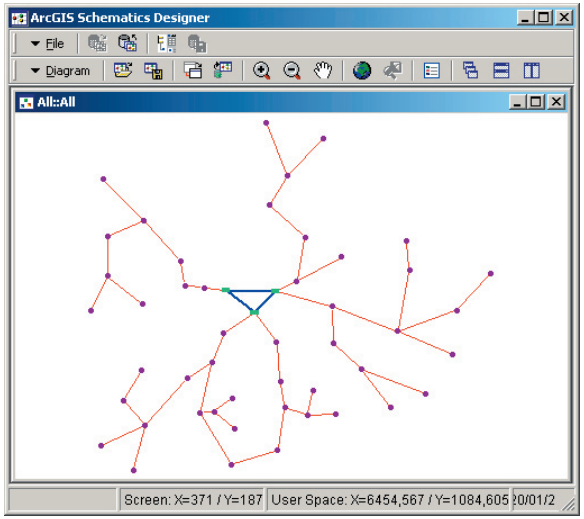
7. Right-click the FeederNodes element type entry and click Create Attribute.
8. On the Create Element Type Attribute dialog box, from Name drop-down list, click InitialYPosition, and from the Type drop-down list click Geometry Attribute.
9. Click OK.
10. On the InitialYPosition attribute's General tab, from the Type drop-down list, click Y-coordinate value.
11. From the Field Name drop-down list, click SHAPE and validate.

The FeederNodes nodes should now be displayed at their geographic location.

You need to repeat these 11 steps for creating the same InitialXPosition and InitialYPosition attributes for the StationNodes element type so these nodes also display at their geographic location.

Click Save to save the specified parameters.

You can now switch to Run mode and test the All diagram display. It should appear as follows:



For the link element types, there is also a specific attribute InitialListPoints that can be set so ArcGIS Schematics automatically displays vertices along links.

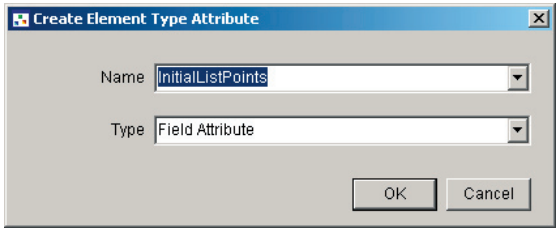
To be interpreted correctly, the values returned by the InitialListPoints attribute must be strings formatted as follows:

N;X1;Y1;…;Xi;Yi;...;XN;YN where:

- N is the number of vertices.
- Xi and Yi are the coordinates of point i.
- The ";" character works as a separator.

In the Schematics_Designer_Tutorial database, the ListPoints field containing such string values is available from the LV_Line links table. To use this database field, the InitialListPoints specific attribute must be created as follows:

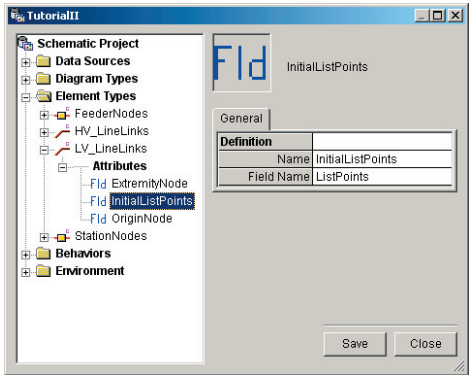
12. In the Designer tree, right-click the LV_LineLinks element type entry and click Create Attribute. The Create Element Type Attribute dialog box opens.
13. From the Name drop-down list, click InitialListPoints.
14. In the Type drop-down list, make sure that the Field Attribute value is selected.



15. Click OK.
16. Then, from the Identifier Editor dialog box that opens and from the Field list, click the ListPoints field and validate by clicking the Down arrow button.
17. Click OK.

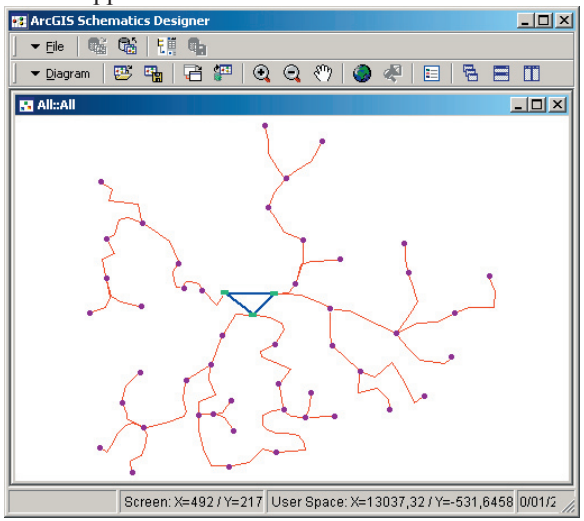
The Identifier Editor dialog box closes.

At the end of these steps, the InitialListPoints attribute's General tab should display as follows:



18. Click Save to save the specified parameters.

You can now switch to Run mode and test the All diagram display. It should appear as follows:



Exercise 4: Symbology filters and labeling for schematic elements

In this exercise, you will learn how to label and symbolize the schematic elements contained in your schematic diagram according to data stored in the database. You will become familiar with property creation and use.

Properties can be created and defined for all types of schematic elements but only on one condition—the element type must be an element group. Properties are used to highlight the characteristics of a group of schematic elements. When they are enabled, they have a direct impact on the graphic representation of a network. The overall state of a network is determined by the values taken by the existing set of properties at a given time. The values taken by properties are often field values stored in the database table that the schematic element type query is based on. In this case, an attribute related to the desired database field must be defined and associated with the property. When this association is done, the schematic element's database field value must be interpreted by the property to determine each element's appearance.

This exercise is organized into three parts:

- In the first part, you will create textual properties for labeling your schematic nodes and links. You will also learn about how to create and use flags to display these labels.
- The second part's purpose is to create discrete properties. You will also learn about how to create a composed CGM symbol to display a node element type and use this composed symbol to specify interesting subsymbols' effects to represent your discrete filters.
- During the third part, you will learn about bounded properties. You will also learn how to create pattern models that can be an interesting alternative to the basic effects usually used for displaying schematic links.

Displaying labels for StationNodes and LV_LineLinks

In this part, you will create two textual properties:

- The Label1 property will display the OBJECTID of each link in blue, centered below each link.
- 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.

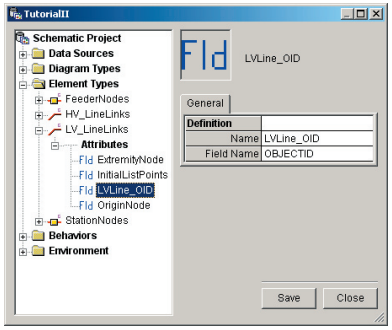
Then, in a second step, you will create a flag model and will learn how flags can be attached on schematic nodes to display the Label2 property labels.

Displaying a label below the schematic links

The Label1 property will use the OBJECTID field returned by the LV_LineLinks query. Begin by creating an attribute corresponding to this OBJECTID field:

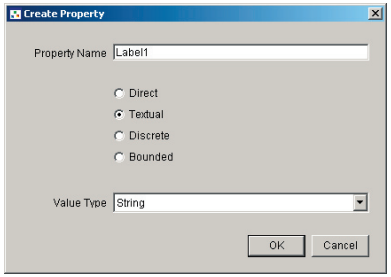
1. Right-click the LV_LineLinks entry in the Designer tree and click Create Attribute.
2. In the Create Element Type Attribute dialog box,type the name of the new attribute in the Name field (LVLine_OID in this example), click Field Attribute from the Type drop-down list, and click OK to validate.
3. On the Identifier Editor dialog box that opens, click the OBJECTID field from the Fields list, click the Down arrow button, and click OK.

The Identifier Editor and Create Element Type Attribute dialog boxes close. The General tab related to the newly created LVLine_OID attribute should display as follows:



Now, you must create the textual property that will manage this attribute display:

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

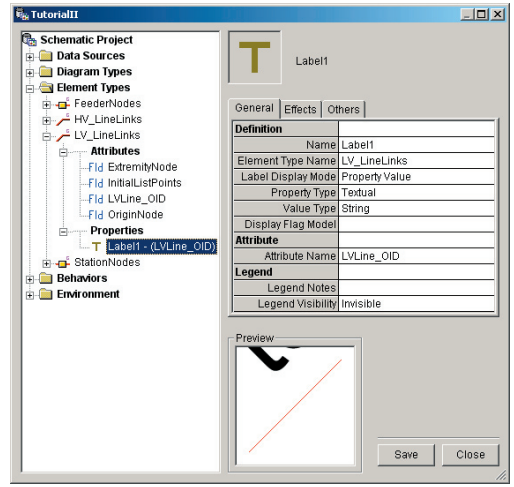


6. Since the default parameters set in this dialog box (Textual property type and String value type) are those you need, click OK.
   The Create Property dialog box closes.

Under the LV_LineLinks 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 LVLine_OID attribute with the newly created property:

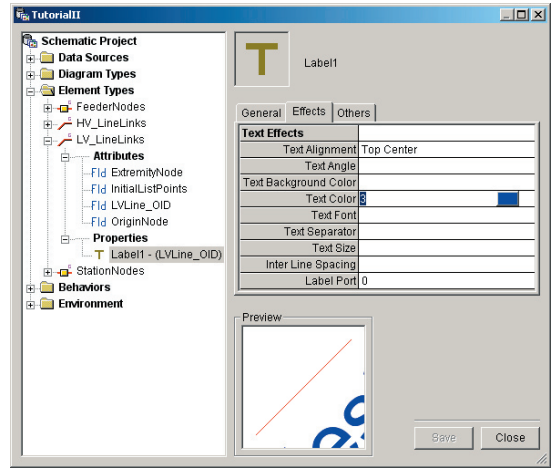
7. Click the General tab.
8. From the Attribute Name drop-down list, click LVLine_OID.
9. From the Legend Visibility drop-down list, click Invisible so this label doesn't display in the Legend subwindow and in the ArcMap TOC (when the diagram will be open in the ArcMap environment).

At the end of these steps, the Label1 textual property's General tab should display as follows:



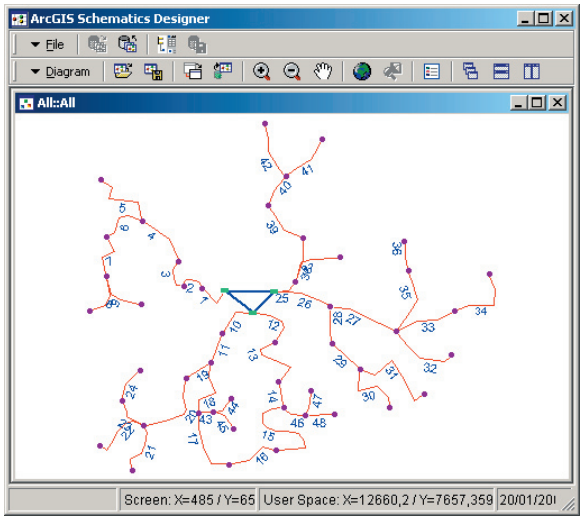
Now, you need to specify how the label will display. The label must display in blue, centered below each link.

10. Click the Effects tab where the text effects are assembled.
11. From the Text Alignment drop-down list, click Top Center.
12. Click the Text Color field and click the color box that displays to launch the Color Editor. Choose the desired blue color and validate.



13. Click Save to save the specified parameters.

If you want to test the Label1 property definition result, click Close and open the All diagram. The result should look similar to the following screen shot:



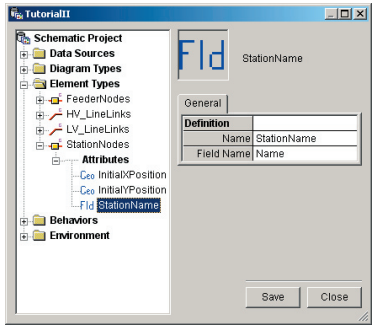
Switch to Design mode.

Displaying a label for the schematic nodes

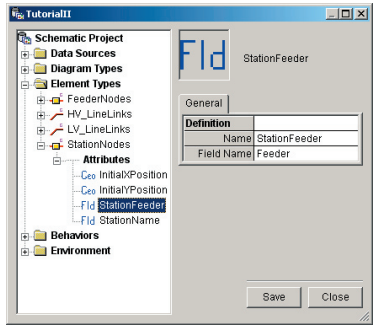
The Label2 property must use the Name and Feeder field returned by the query specified for the StationNodes element type as formatted parameters to display a label formatted as follows: Name (F-Feeder). You must begin by creating the two attribute fields related to the station's Name and station's Feeder database field before building the final formatted attribute that will be displayed.

1. Right-click the StationNodes element type and click Create Attribute.
2. From the Create Element Type Attribute dialog box that opens, type StationName in the Name field, click Field Attribute from the Type drop-down list, and click OK to validate.
3. From the Identifier Editor dialog box that opens, click the Name item from the Field's list, click the Down arrow, and validate.

The Identifier Editor and Create Element Type Attribute dialog boxes close.



4. Proceed in the same way to create the StationFeeder attribute field related to the Station table's Feeder field.



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

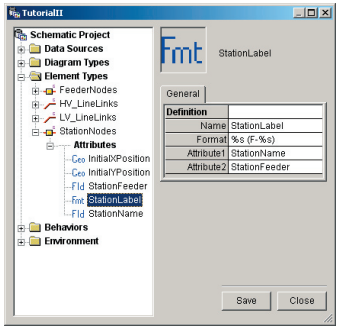
5. Right-click the Attributes entry related to the StationNodes element type and click Create.
6. From the Create Element Type Attribute dialog box, in the Name field, type “StationLabel”; in the Type drop-down list, click Formatted Attribute and click OK.
7. From the General tab related to the newly created attribute, you then 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 one 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.
8. Press Enter to validate the Format parameter.

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

9. From the Attribute1 drop-down list, click StationName.
10. From the Attribute2 drop-down list, click StationFeeder.

At the end of these steps, the StationLabel attribute’s General tab should display as follows:

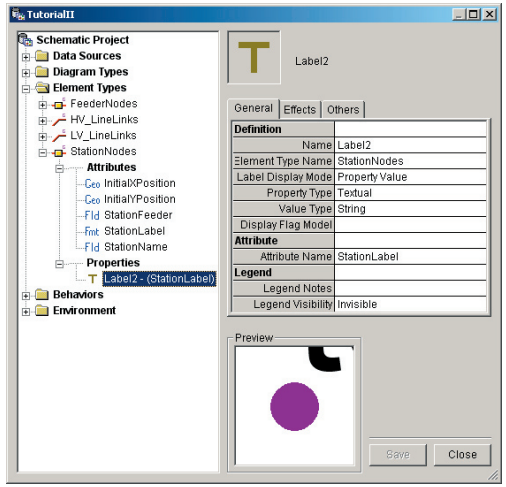


As the label values you want to display for your station nodes are defined, you can now create the textual property that will manage this information display.

11. Right-click the StationNodes element type entry and click Create Property.
12. In the Create Property dialog box that opens, type “Label2” in the Name field and click OK.

Under the StationNodes 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 StationLabel attribute with the newly created property.

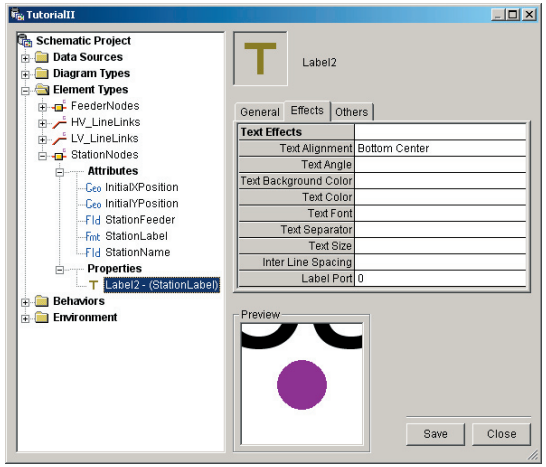
13. Click the General tab.
14. From the Attribute Name drop-down list, click StationLabel.
15. From the Legend Visibility drop-down list, click Invisible so this label doesn’t display in the Legend subwindow and in the ArcMap TOC (when the diagram will be open in the ArcMap environment).



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

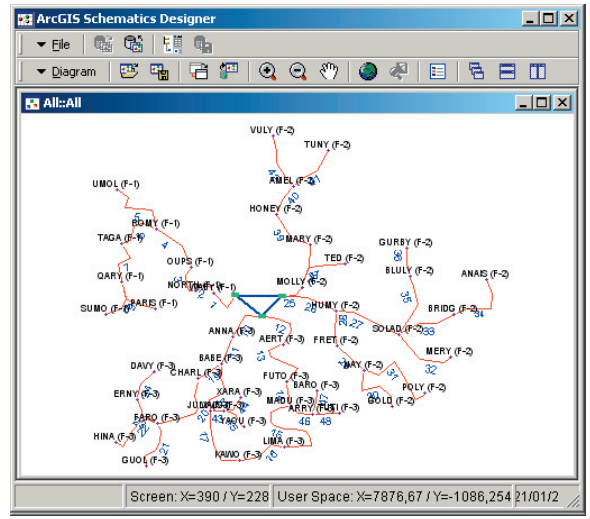
16. From the Label2 property's Effect tab, click the Text Alignment drop-down list and click Bottom Center.

Note that 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.



17. 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.

18. Switch to Design mode.
19. In the Designer tree, click Schematic Project and click the Effects tab.
20. Type "0.2" in the Text Size field.
21. Save, close the Designer Editor window, and test the network display again.

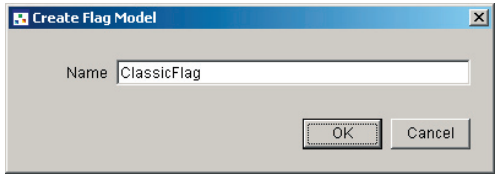
    When the diagram is at full extent, the labels are invisible, but when zooming in on a set of elements, the labels appear.

Displaying labels in flags

In this part, you will have to modify the Label2 property defined for the StationNodes element type so this property's labels are displayed through a flag attached to each station.

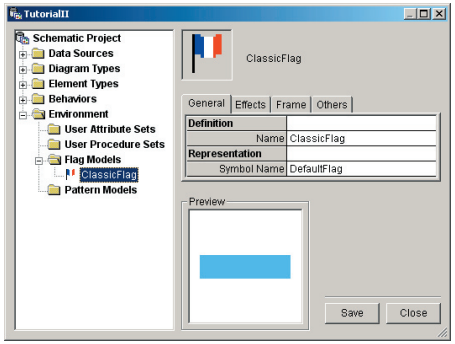
Begin this step by creating the flag model that will implement the flags into which the labels will display.

1. In the Designer tree, expand Environment, right-click Flag Models, and click Create.
2. In the Create Flag Model dialog box that opens, type a name for the new flag model (ClassicFlag in this example) in the Name field.



3. Click OK.

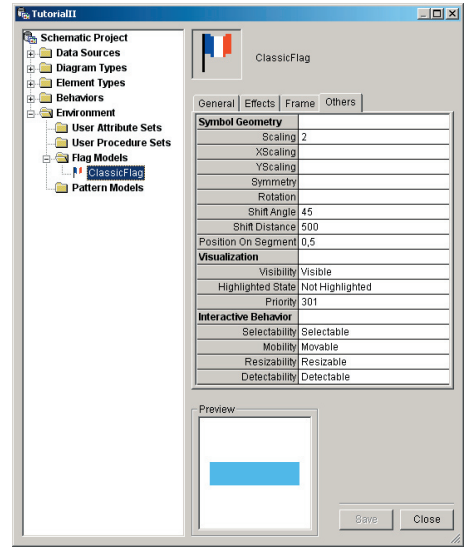
The new flag model entry is automatically created below the Flag Models entry. The Preview window shows the symbol that will be used by default to represent all flags of this type. The default flag symbol (named DefaultFlag) you defined at the Schematic Project level in exercise 2 displays in this window:



Now, you need to specify the flag model parameters:

4. On the General tab, change the Symbol Name parameter by creating a new CGM symbol, if desired.
5. Click the Others tab, where parameters related to the flag geometry and visualization are available.
6. Type "2" for the Scaling parameter.
7. Specify how the flags will be displayed regarding the elements they attach to: type "45" for Shift Angle and "500" for Shift Distance.
8. Save your parameters definition.

At the end of these steps, the flag model's Others tab should display as follows:

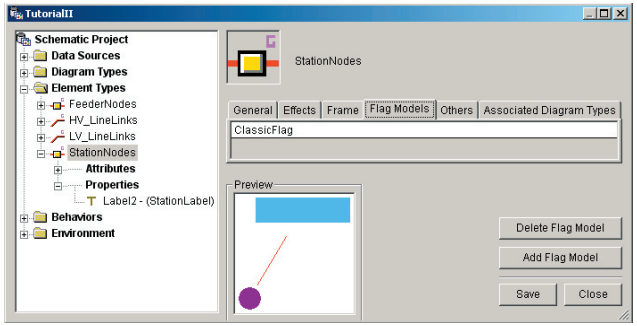


The purpose now is to modify the Label2 property so this new flag model is used to display the property label.

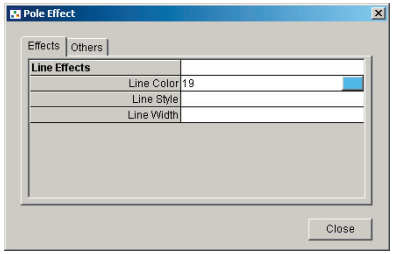
9. In the Designer tree, click the Label2 property entry and click the General tab.
10. From the Display Flag Model drop-down list, click ClassicFlag.
11. Click the Text Alignment parameter from the Effects tab and click Half Center so labels display centered on their attached flags.

Selecting ClassicFlag for the Display Flag Model drop-down list in the property's General tab caused the ClassicFlag flag model to be associated with the StationNodes element type group; a new Flag Models tab has been automatically created for this element type.

From this new FlagModels tab, you can specify parameters for the pole that will connect each flag to its attached element:



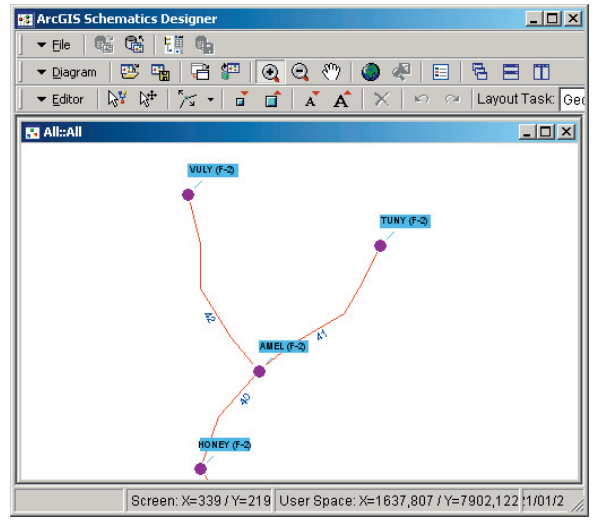
12. Click StationNodes element type in the Designer tree, and click the newly created Flag Models tab.
13. Click the ClassicFlag flag model line that has just been automatically added to the Flag Models tab and click the Ellipsis button.
    The Pole Effect dialog box opens.
14. Click the Effects tab, click the Line Color parameter, and specify the color you want using the Color Editor.



15. Click Close. The Pole Effect dialog box closes.

16. Click Save. Switch to Run mode and test the flags display.

After zooming in on a part of the All schematic diagram content, the result should be similar to the following screen shot:



Creating discrete properties for StationNodes and LV_ LineLinks

In this part, you will learn a discrete property is a property with specified values that belong to a finite set of numbers or of 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 low-voltage links display according to the Category database field values. Each link is an S, M, or B category (for Small, Medium, or Big, respectively). A discrete property will be created, and display rules will be defined for the discrete filters based on these categories.
- Then, after defining a composed CGM symbol to represent the station nodes, you will learn how to define subsymbol effects to symbolize the station nodes display according to the Type database field values. Since this field 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 field returned by the LV_LineLinks element type query stores the category of each link. Each link is an S, M, or B category (for Small, Medium, or Big, respectively).
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_LineLinks according to the Category field as follows:

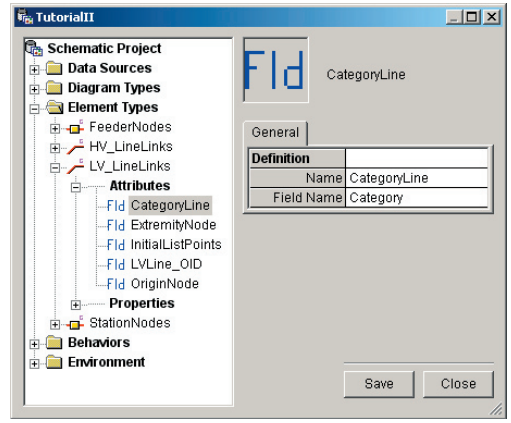
Category Value	Filter Graphic Effect
S	Line Width: 1.0
M	Line Width: 2.0
B	Line Width: 4.0

Since the DisplayByCategory property will use the Category field values returned by the query, you must begin by creating an attribute field corresponding to this field.

1. Expand the Element Types entry, right-click LV_LineLinks, and click Create Attribute.
2. In the Create Element Type Attribute dialog box that opens, type a name for the attribute (CategoryLine in this example) in the Name field, click Field Attribute value from the Type drop-down list and click OK.
3. On the Identifier Editor dialog box, click Category from the Fields list, click the Down arrow to validate this field, and click OK.

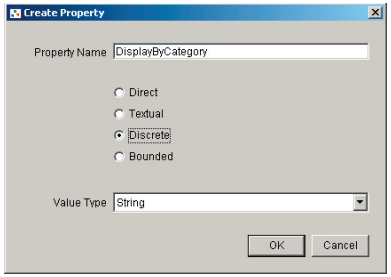
The Identifier Editor and Create Element Type Attribute dialog boxes close.

At the end of these steps, the CategoryLine attribute's General tab should display as follows:



Now, you will create the DisplayByCategory discrete property that will display each LV_LineLinks link according to the CategoryLine attribute value.

4. Right-click the Properties entry displayed below the LV_LineLinks element type and click Create.
5. In the Create Property dialog box that opens, in the Property Name field, type "DisplayByCategory", check the Discrete option, and keep String for Value Type.



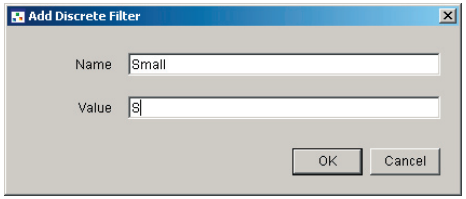
6. Click OK.
   The Create Property dialog box closes.

Now, you have to associate this new property with the CategoryLine attribute.

7. From the DisplayByCategory property's General tab, from the Attribute Name drop-down list, click CategoryLine.

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

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

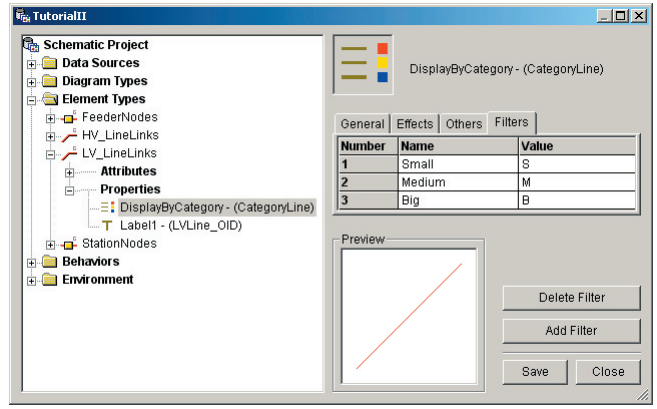


10. Click OK to validate the discrete filter definition.

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

11. Repeat steps 9 through 10 to create two new discrete filters for the M and B values of the Category field database.

At the end of these steps, the Filters tab should display as follows:

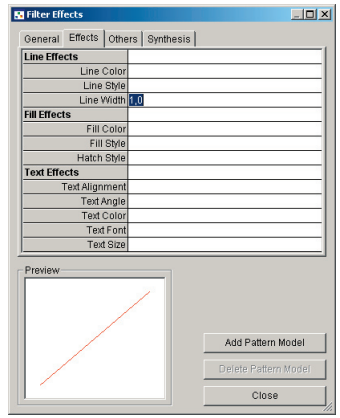


Now, you need to define the graphical effects related to each discrete filter.

12. Click the Filters tab, click the first discrete filter parameter line (Small filter), and click the Ellipsis button that displays.

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

13. On the Filter Effects dialog box, click the Effects tab and type "1,0" in the Line Width field.



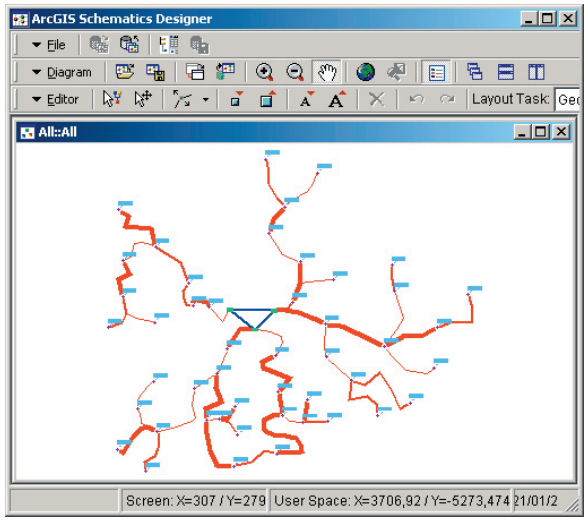
14. Click Close to validate this parameter and close the Filter Effects dialog box.

15. Repeat steps 12 through 14 to define the effects related to Medium and Big discrete filters as follows:

- For the Line Width parameter field related to the Medium filter, type "2,0".
- For the Line Width parameter field related to the Big filter, type "4,0".

16. Click Save.

If you test the All diagram content display, it should appear as follows:



For a better understanding of the other exercises, you will temporarily disable the display of the ClassicFlag.

17. Click the Label2 entry and deselect ClassicFlag from the Display Flag Model field.
18. Click the StationNodes element type in the Designer tree and click the Flag Models tab.
19. Click the line corresponding to the ClassicFlag flag model and click Delete Flag Model
20. Click Save.

Symbolizing the schematic nodes according to their type

The Type field returned by the StationNodes element type query stores the type of each station. Each station node is either an A, B, C, D, E, or F type. The purpose of this section is to create the DisplayByType discrete property 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 have created during the second exercise for which you will add two subsymbols. This composite symbol will be used by default to represent the StationNodes elements. Then, 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 StationNodes element type is represented by the default node symbol, specify this new composed symbol for representing StationNodes.

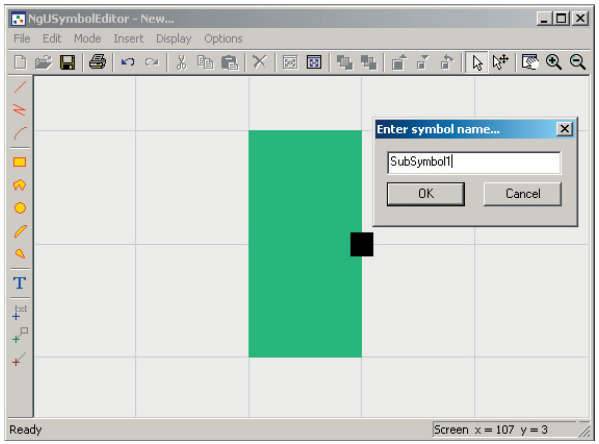
1. Expand the Element Types entry, and in the Designer tree, click StationNodes.
2. Click the StationNodes' General tab, click the Symbol Name field, and click the NgUSymbolEditor button that displays to launch the CGM Symbol Editor and Vector Drawing tool.

Begin with the creation of the first subsymbol:

3. Click the Insert Rectangle button and draw a green rectangle as in the following screen shot.

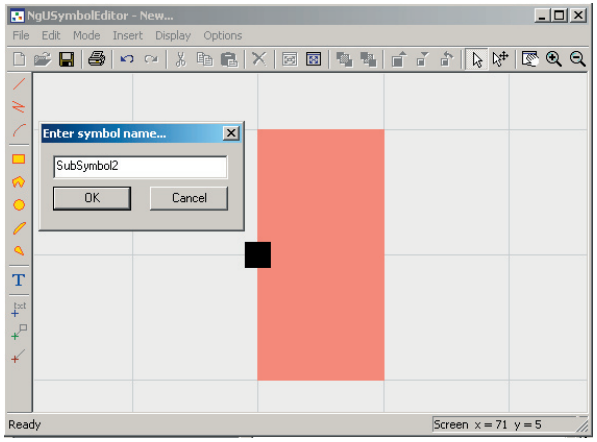
Note: To change the rectangle color fill, double-click the newly created rectangle to open the Graphic Attributes dialog box; click the Fill Attributes tab and change the Color parameter.

Click File, click Save, type a name for the first subsymbol (SubSymbol1 for example), and validate.



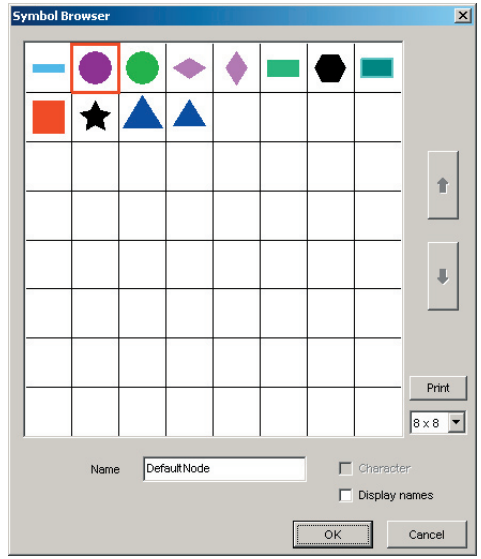
Then, create the second subsymbol.

4. Click File and click New. Click the Insert Rectangle button and draw a pink rectangle as in the following screen shot. Save this new symbol (as SubSymbol2 for example).



Now, you will create the composite CGM symbol. This symbol will be based on the DefaultNode CGM symbol built during the second exercise and will combine the two subsymbols you have just created.

5. Click File and click New.
6. Click File and click Import.
7. From the Symbol Browser dialog box that opens, select the DefaultNode CGM symbol.



8. Click OK. The Symbol Browser dialog box closes.
9. From the Import dialog box that displays, keep the Import graphics option checked and click OK. The symbol related to the DefaultNode CGM symbol displays.

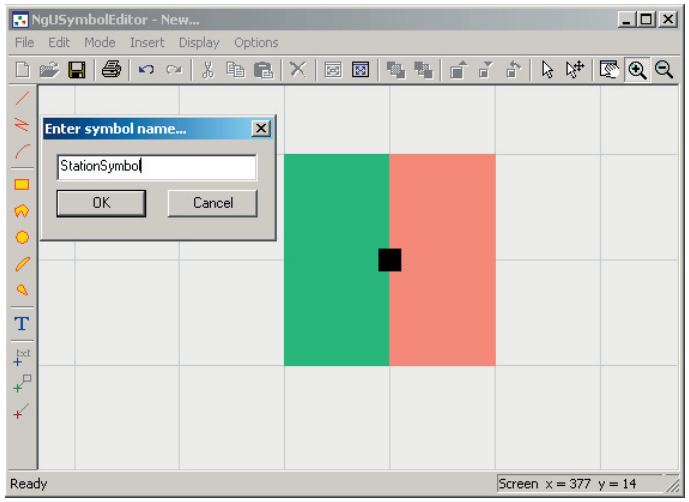
You must now insert the first subsymbol you have created.

10. Click Insert, point to SubSymbol, and click Internal SubSymbol.
11. From the Symbol Browser dialog box that opens, click SubSymbol1.
12. Click OK.

The SubSymbol1 inserts.

13. Repeat steps 10 through 12 to insert the second subsymbol.

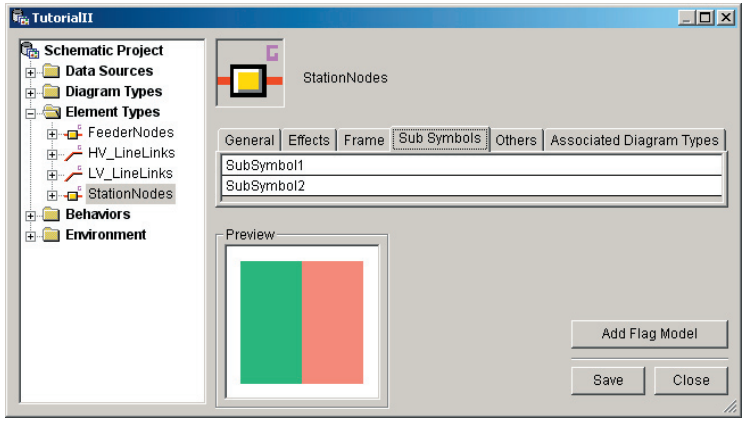
14. Save your composed CGM symbol.



15. Exit the NgUSymbolEditor drawing tool.

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

The Sub Symbols tab, which contains all subsymbols that comprise this symbol, appears among the other tabs.



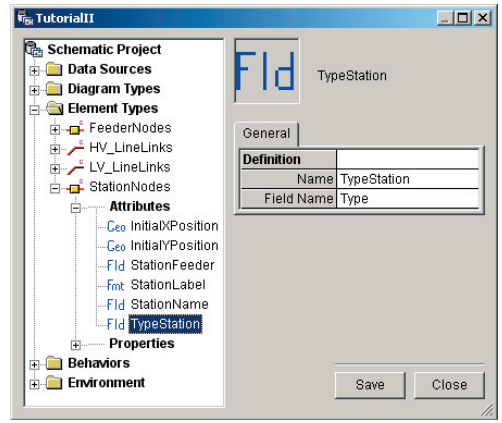
Now, you will create the DisplayByType discrete property.

As this discrete property will be based on the Type field values stored in the Station table, begin by creating the attribute field related to this database field.

16. Right-click the StationNodes element type entry and click Create Attribute.
17. On the Create Element Type Attribute dialog box that opens, in the Name field, type a name for the attribute (TypeStation in this example), from the Type drop-down list, click Field Attribute value and click OK.
18. On the Identifier Editor dialog box that opens, click Type from the Fields list, click the Down arrow to validate this field, and click OK.

The Identifier Editor and the Create Element Type Attribute dialog boxes close.

At the end of these steps, the TypeStation General tab should display as follows:



Now, you will create the DisplayByType discrete property that will display each StationNodes node according to the TypeStation attribute value.

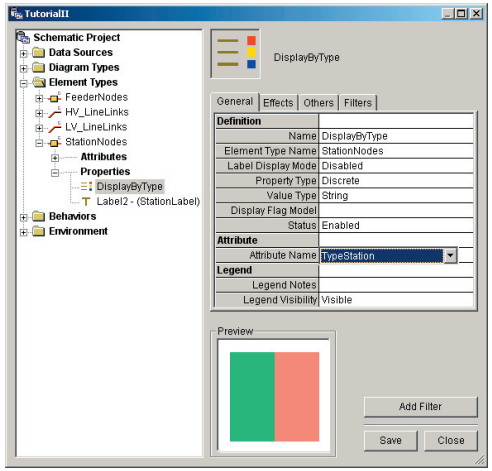
19. Below the StationNodes element type, right-click Properties and click Create.

20. In the Create Property dialog box that opens, in the Property Name field, type “DisplayByType”, check the Discrete option, keep String for Value Type, and validate.

The Create Property dialog box closes.

21. You must now associate the property with the TypeStation attribute.

From the DisplayByType General tab, from the Attribute Name drop-down list, click TypeStation.



Now, create the discrete filters related to each Type field’s value.

22. Click the Add Filter button displayed on the lower right corner of the Designer Editor window.

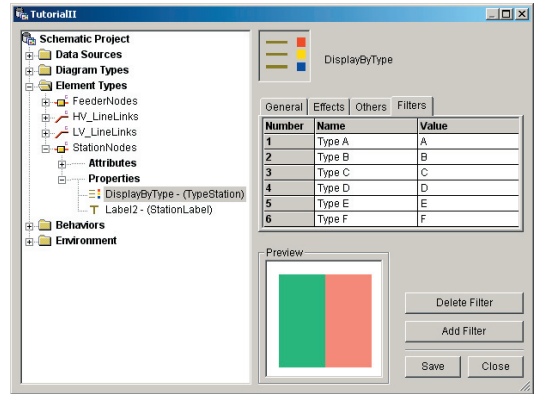
23. From the Add Discrete Filter dialog box that opens, type a name for the discrete filter (Type A for example) and specify the first value for the discrete filter (A in this example). This value is one of the values taken by the Type field stored in the Station table.

24. Click OK to validate the discrete filter definition.

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.

At the end of these steps, the Filters tab should display as follows:



Now, you will define the graphic effects that will represent each discrete filter.

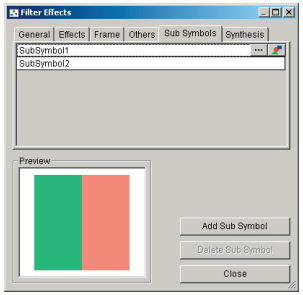
To define graphical effects related to a filter, you will need to click the Filters tab and choose the desired discrete filter parameter. Then, by clicking the Ellipsis button displayed at the end of this parameter line, the Filter Effects form will automatically open.

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

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

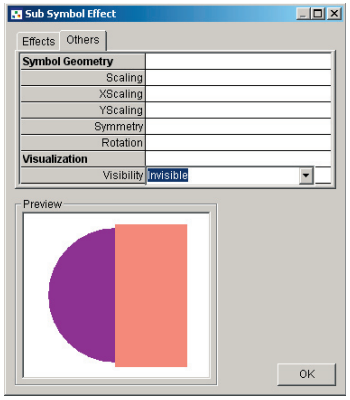
Start with the Type A filter to set the first subsymbol as invisible.

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



The Sub Symbol Effect dialog box related to SubSymbol1 opens.

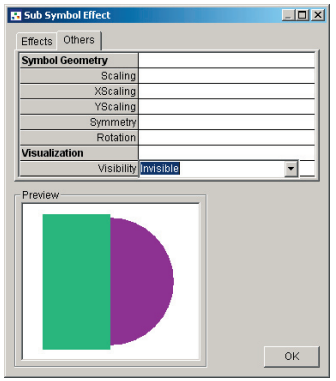
29. Click the Others tab, and from the Visibility drop-down list click Invisible.



30. Click OK. The Sub Symbol Effect dialog box closes.
31. Close the Filter Effects dialog box.

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

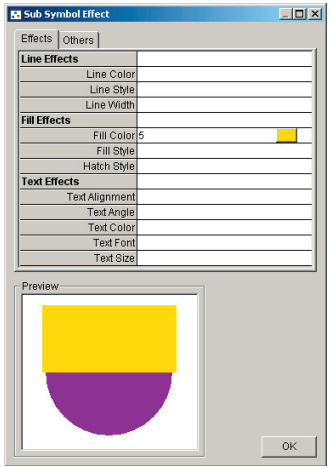
32. On the Filters tab, click the Type B discrete filter and click the Ellipsis button.
33. From the Filter Effects dialog box, click the Sub Symbols tab.
34. Double-click the line corresponding to SubSymbol2 and click the Ellipsis button that displays.
35. Click the Others tab, and from the Visibility drop-down list click Invisible.



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. On the Others tab, from the Visibility drop-down list click invisible and validate so the subsymbol effect related to the first symbol closes.
41. Then, from the Sub Symbols tab, open the Sub Symbol Effect dialog box related to SubSymbol2.
42. On the Others tab, in the Rotation field, type “90”.
43. Click the Effects tab. Click the Fill Color field and set the color you want.

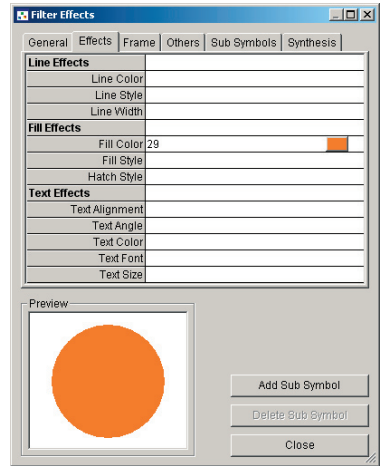


44. Click OK to close the Sub Symbol Effect dialog box.
45. Close the Filter Effects dialog box.

Now, 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. On the Others tab, from the Visibility drop-down list, click invisible and validate so the subsymbol effect related to the first symbol closes.
49. Repeat steps 47 and 48 to disable the visibility for SubSymbol2.
50. From the Filter Effects dialog box click the Effects tab. Click the Fill Color parameter and set the color you want.

The Preview subwindow on the Filter Effects dialog box should display similarly to the following screen shot:

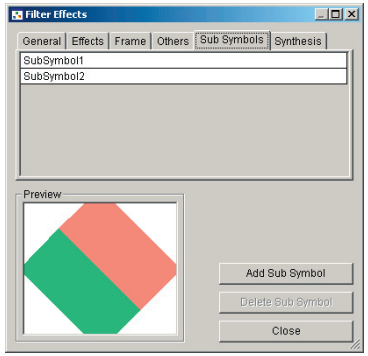


51. Close the Filter Effects dialog box.

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. On the Others tab, type “45” for the Rotation parameter and validate 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.

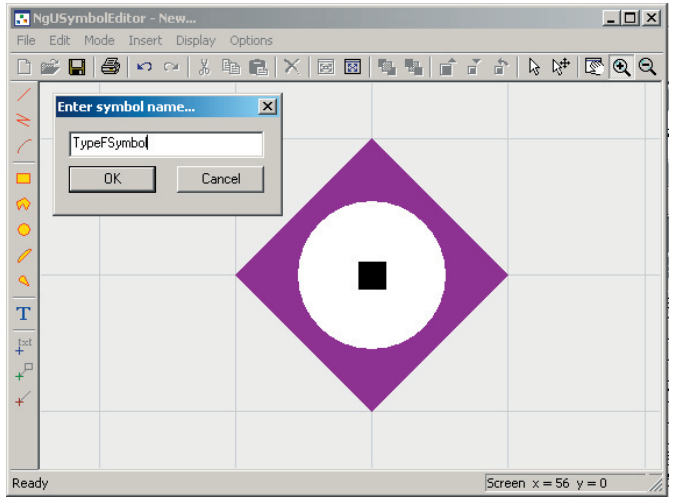
The Preview subwindow in the Filter Effects dialog box should display similarly to the following screen shot:



56. Close the Filter Effects dialog box.

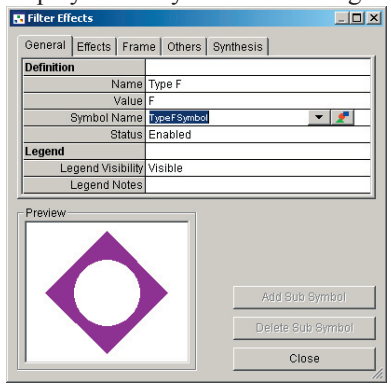
For the Type F discrete filter, you must create a new CGM symbol.

57. Open the Filter Effects dialog box corresponding to the last discrete filter (Type F).
58. On the General tab, click the Symbol Name field and click the NgUSymbolEditor button that displays to launch the CGM Symbol Editor and Vector Drawing tool.
59. Draw a new CGM symbol and save it.



60. Exit the NgUSymbolEditor.

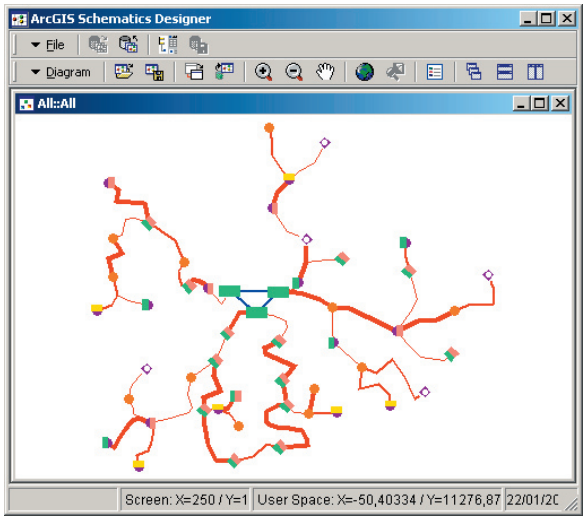
The Symbol Name field on the Filter Effects dialog box automatically fills with the new CGM symbol name. The Preview subwindow should display similarly to the following screen shot:



61. Close the Filter Effects dialog box.

62. Save the discrete property definition parameters.

If you want to test the impact of the DisplayByType property on the schematic StationNodes elements, click Close, switch to Run mode, and open the All diagram. It should appear as follows:



Creating bounded properties for StationNodes and LV_LineLinks

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.

Two bounded properties will be created:

- You will first animate the station display according to the Capacity database field 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.
- Then, you will create pattern models. A pattern is a symbol or a text that is repeated one or several times along a link path route. Displaying patterns is an interesting alternative to the basic line styles you can set for links. Using pattern models to represent bounded filters, you will animate the network links according to the Rate database field values.

Modifying the schematic station sizes according to their capacities

The Capacity field returned by the StationNodes query stores the capacity of each station as an integer value.

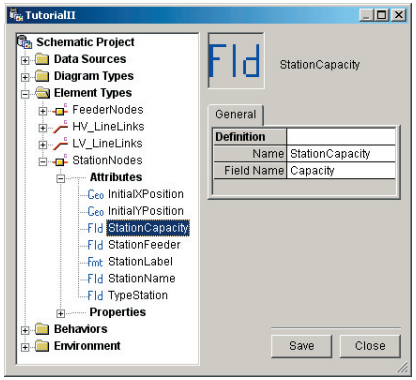
In this section, you will need to represent this information graphically. You will create the DisplayByCapacity bounded property composed of four bounded filters with each filter corresponding to one range of values. This property will display the nodes according to the Capacity field values as follows:

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

Since the DisplayByCapacity property will use the Capacity field returned by the query, you must begin by creating an attribute corresponding to this field.

1. Right-click the Attributes entry displayed under the StationNodes element type and click Create.
2. On the Create Element Type Attribute dialog box that opens, in the Name field type the name that will be used to reference this attribute (StationCapacity for example); from the Type drop-down list, click Field Attribute and validate.
3. Use the Identifier Editor to select the Capacity field from the Fields list and validate it.

At the end of these steps, the StationCapacity attribute should display as follows:

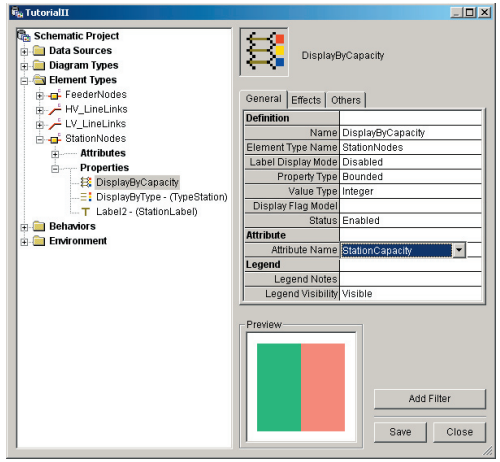


The purpose now is to create the DisplayByCapacity bounded property that will display each station according to the StationCapacity attribute values.

4. In the Designer tree, under StationNodes, right-click Properties and click Create.
5. In the Create Property dialog box that opens, type “DisplayByCapacity” in the Name field.
6. Check the Bounded option.
7. From the Value Type drop-down list click Integer.
8. Click OK.

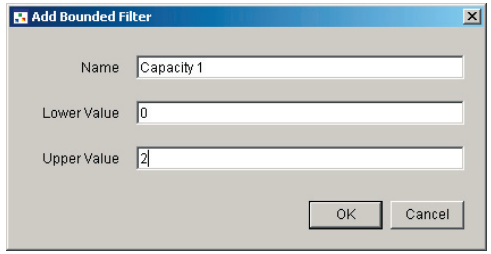
The new DisplayByCapacity property is automatically referenced below the Properties entry.

9. In the General tab, associate the property with the StationCapacity attribute by clicking StationCapacity from the Attribute Name drop-down list.



Now, you need to create the bounded filters.

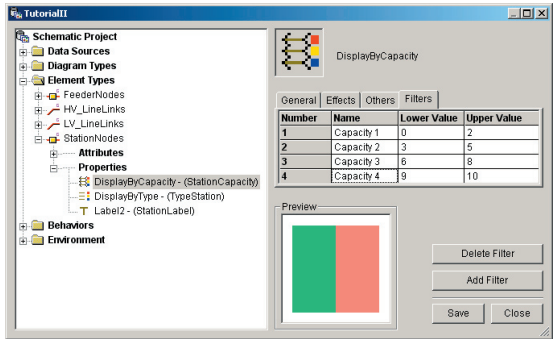
10. Click the Add Filter button on the lower right corner of the Designer Editor window.
11. From the Add Bounded Filter dialog box that opens, in the Name field, type a name for the first bounded filter (Capacity 1 for example) and set the lower and upper values (0 and 2 in this exercise) that will define this first bounded range.



12. Click OK.
    The Add Bounded Filter dialog box closes.
13. Repeat steps 10 through 12 to create three new filters corresponding to:

- The 3 to 5 value range (Capacity 2)
- The 6 to 8 value range (Capacity 3)
- The 9 to 10 value range (Capacity 4)

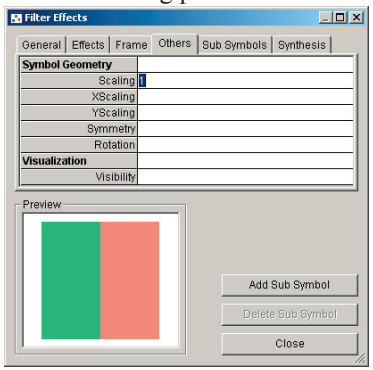
At the end of this step, the Filters tab should look like the following graphic:



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

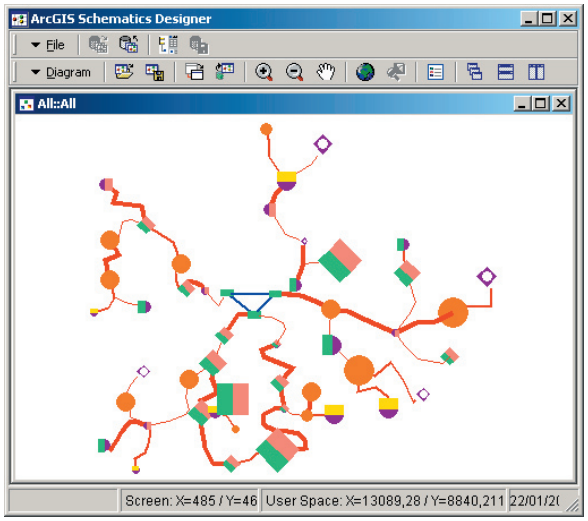
Start with the Capacity 1 bounded filter that must set the station scaling factor to 1.

14. From the Filters tab, click the Capacity 1 bounded filter and click the Ellipsis button that displays.
15. On the Filter Effects dialog box, click the Others tab and type “1” for the Scaling parameter.



16. Click Close to close the Filter Effects dialog box.
17. For defining the graphic effects related to Capacity 2, 3, and 4, repeat steps 14 through 16 to set the Scaling parameter with values of 2, 3, and 5, respectively.
18. Click Save.

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:



Using pattern models to animate the links according to the Rate field values

In this step, you will create two pattern models that you will use afterward to display the links according to a new bounded property with values that will be based on the Rate field stored in the LV_Line table.

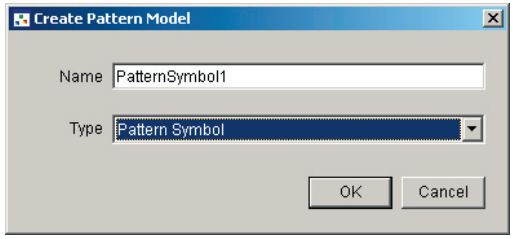
The first PatternSymbol1 model you will create is a pattern model symbol that will display an arrow placed on the middle of each link path route.

The second PatternText1 model will be a pattern model text that displays the X character along each link path route. The rules to repeat the pattern along the segment are as follows:

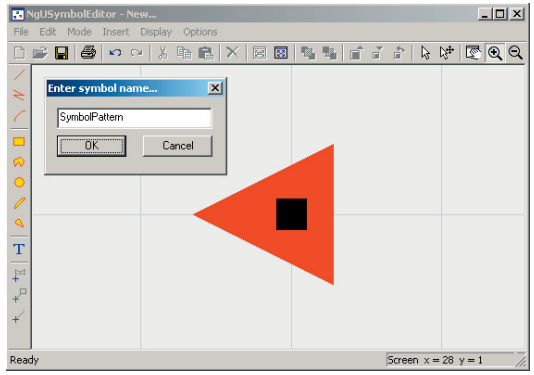
- Spacing between items is 5.0.
- First item is 3.0 units away from link origin (Shift Distance).

The steps required to create the PatternSymbol1 model are detailed below:

1. Under the Environment entry, right-click the Pattern Models entry displayed and click Create.
2. From the Create Pattern Model dialog box that opens, type the name that will be used to reference your first pattern model (PatternSymbol1 in the example).
3. From the Type drop-down list, click Pattern Symbol.

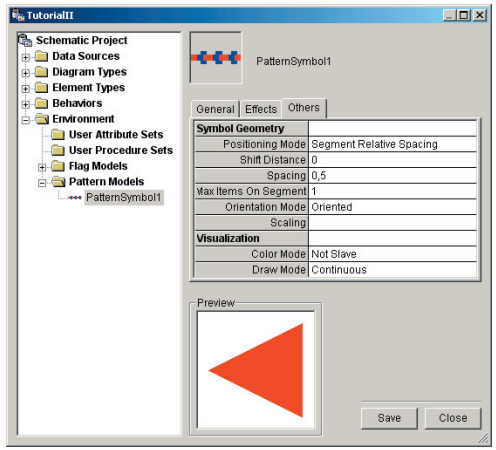


4. Click OK.
5. From the General tab, specify the CGM symbol that will represent each pattern. After you click the Symbol Name field, click the NgUSymbolEditor button displayed to launch the NgUSymbolEditor.
6. Draw the new CGM file representing the desired symbol and save it.



7. Close the NgUSymbolEditor.
8. The parameters needed to define the rules that will be used to repeat the patterns along the segment are those available from the pattern model Others tab. Click the Others tab.
9. The default values set for the Positioning Mode, Shift Distance, Spacing, and Max Items On Segment parameters are those you need to display a pattern on the middle of each link path route.

   Your first pattern model definition is finished.

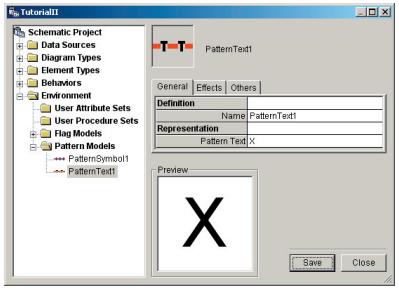


The second PatternText1 pattern model you must create is a pattern model text that will display the X character along each link path route.

The pattern color should automatically change according to the color of the link in which it will be displayed. The rules to repeat the pattern along the segment are as follows:

- Spacing between items is 5.0.
- First item is 3.0 units away from link origin (Shift Distance).

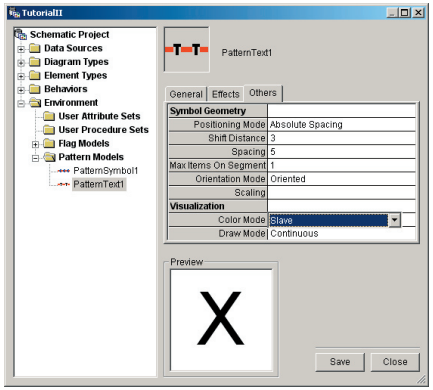
10. Right-click the Pattern Models entry and click Create to open the Create Pattern Model dialog box.
11. In the Name field from the Type drop-down list, type the name that will be used to reference your second pattern model (PatternText1, for example), click Pattern Text value , and click OK.
12. Below the Pattern Models entry, click the new pattern model and on the General tab type “X” in the Pattern Text field, as in the following screen shot:



13. Click the Effects tab, and from the Text Alignment parameter field click Half Left.

Now, you must define the rules that will be used to repeat the patterns along the segment. The pattern model must implement five items along the link path route, with the first item 3 units away from the link origin:

14. Click the Others tab, and from the Positioning Mode drop-down list click Absolute Spacing.
15. For the Shift Distance parameter field, type “3”.
16. For the Spacing parameter field, type “5”.
17. Click Slave from the Color Mode drop-down list so the pattern color automatically changes according to the link color.



The second pattern model definition is finished.

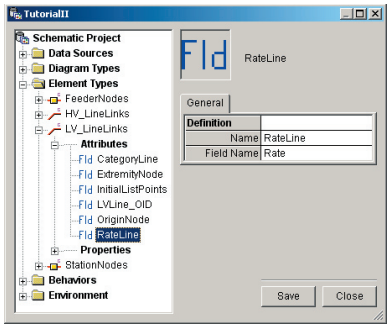
18. Click Save.

The purpose now is to create the DisplayByRate bounded property that you will use to animate LV_LineLinks according to the Rate field returned by the LV_LineLinks query. This Rate field contains double values. The property will display the links according to the Rate field as follows:

Rate Lower value	Rate Upper Value	Filter Graphic Effects
0	200,000	Use the PatternSymbol1 pattern model.
200,000	400,000	Display the link in green and use the PatternText1 pattern model.
400,000	600,000	Use the PatternSymbol1 and PatternText1 pattern models.
600,000	800,000	Display the link in pink.
800,000	1,000,000	Display the link in black with the dashed dotted line style.

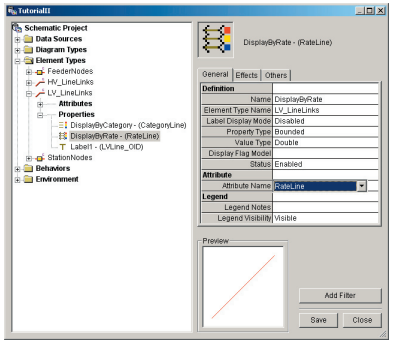
Because the DisplayByRate bounded property will use the Rate field returned by the query to animate the link display, you must create an attribute corresponding to this field.

19. Under the LV_LineLinks element type, right-click the Attributes entry and click Create.
20. From the Create Element Type Attribute dialog box that opens, in the Name field, type “RateLine” and validate.
21. On the Identifier Editor dialog box, click Rate from the Field’s list and validate.



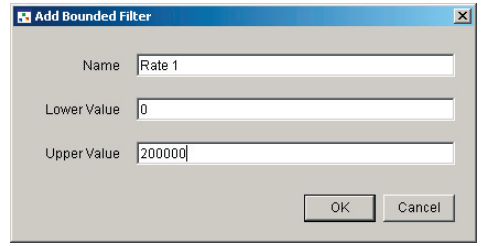
Now, you are going to create the DisplayByRate bounded property that will display each link according to the created RateLine attribute values.

22. Under the LV_LineLinks element type, right-click the Properties entry displayed and click Create.
23. From the Create Property dialog box that opens, in the Name field type “DisplayByRate”, check the Bounded option from the Value Type drop-down list, click Double, and validate.
24. On the property’s General tab from the Attribute Name drop-down list, click RateLine to associate the new DisplayByRate property with the RateLine attribute.

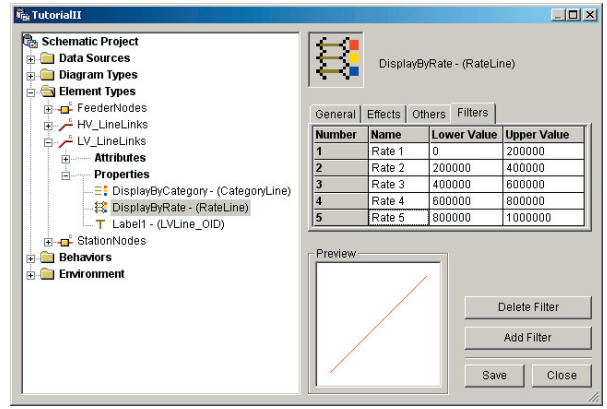


Now, you need to create the five bounded filters that will be used to represent the links according to the DisplayByRate property values.

25. Click the Add Filter button on the lower right of the Designer Editor window to create the property’s first bounded filter.
26. In the Add Bounded Filter dialog box that opens, in the Name field type “Rate 1” and set the lower and the upper values that will define this first bounded range (0 and 200000 respectively).

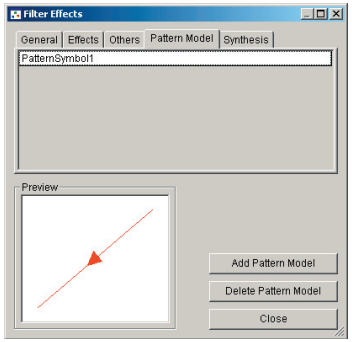


27. Click OK.
28. Repeat steps 26 through 27 to create four other filters corresponding to the value ranges specified in the dialog box above.



29. From the Filters tab, click the Rate 1 bounded filter and click the Ellipsis button that displays to open the Filter Effects dialog box.
30. From the Add Pattern Model dialog box that opens, on the bottom right, click the Add Pattern Model button, click PatternSymbol1, and validate.

The Pattern Model tab automatically displays among the Filter Effects dialog box’s tabs related to the first bounded filter.

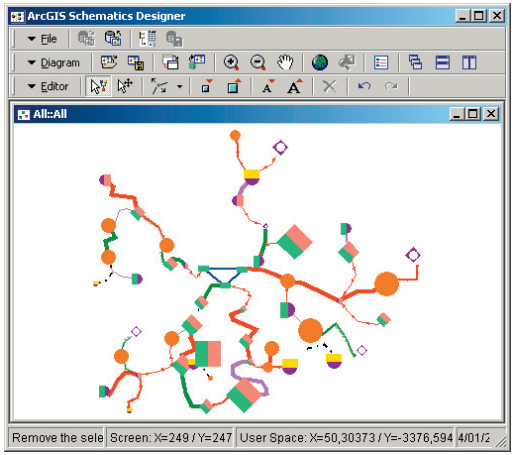


31. Click Close so the Filter Effects dialog box closes.
32. For defining the Rate 2 bounded filter effects, open the Filter Effects dialog box; click Add Pattern Model; and from the Name drop-down list, click PatternText1. Then, click the Effects tab and specify the Line Color parameter as green. Close the Filter Effects dialog box.
33. For the Rate 3 bounded filter, repeat steps 29 through 31 twice: the first time to add the PatternSymbol1 model and the second time to add the PatternText1 model.

For the Rate 4 and Rate 5 bounded filter effects, the graphic effects to specify are all on the Effects tab of the Filter Effects dialog box.

34. Define the graphic effects for the Rate 4 bounded filter by clicking the Filters tab, click the Rate 4 bounded filter, and click the Ellipsis button to open the Filter Effects dialog box. Click the Effects tab and select a pink color for the Line Color parameter. Close the Filter Effects dialog box.
35. Repeat the same operation for the Rate 5 bounded filter: Click the Filters tab, select the Rate 5 bounded filter, and click the Ellipsis button to open the Filter Effects dialog box. Click the Effects tab and choose black for the Line Color parameter. Then from the Line Style drop-down list, click Dashed-Dotted. Close the Filter Effects dialog box.
36. Save the specified parameters.

If you test the All diagram display, it should appear as follows:



Three subnetworks comprise the network stored in the Schematics_Designer _Tutorial database, each one describing a feeder. This exercise will examine the low-voltage part of the network. The purpose is to create new schematic diagram and element types to generate three schematic diagrams, one for each low-voltage subnetwork related to a given feeder. In this way, the Feeder field stored in the Station table and the FromJunctionType and ToJunctionType fields stored in the LV_Line table can be used to filter out the records stored in the database, so you will display three diagrams, one per feeder.

The exercise is organized into three parts:

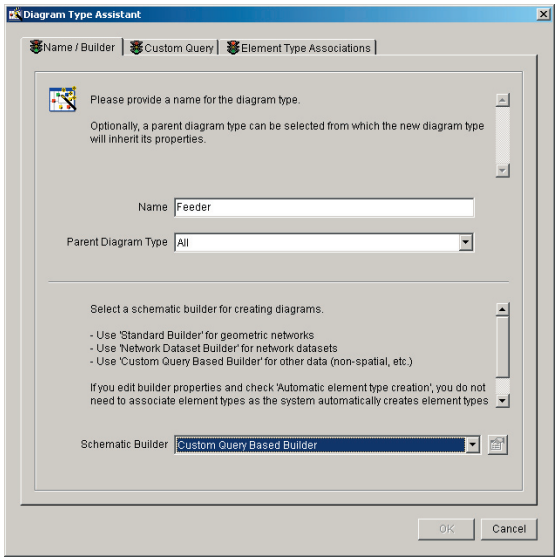
- You will begin by creating a new diagram type that will implement these three diagrams per feeder.
- Then, new node and link element types will be created and associated with this new diagram type. During this part, you will become familiar with the inheritance relationship that can be defined between two schematic element types. The new element type that you create will inherit from the StationNodes and LV_LineLinks element type you created in exercise 3.
- In the last part, you will learn how properties inherited from a parent element type can be redefined at the child element type level.

New with Schematics Designer 9.2, the Diagram Type Assistant guides you in your new diagram types creation and parameters specification, that is, configuring the diagram type parameters and the element types you want to be associated with it.
Until now, you created your diagram type and element types without using the Diagram Type Assistant. In this exercise, you will learn how to use this utility during your schematic project conception. The two first parts will be done using the Diagram Type Assistant.

Creating and configuring a new diagram type to visualize Feeder subnetworks separately using the Diagram Type Assistant

In this step, you will define a new diagram type that will implement the three Feeder subnetworks stored in the database using the Diagram Type Assistant:

1. In the Designer tree, right-click Diagram Types and click Diagram Type Assistant.
2. From the Diagram Type Assistant dialog box that opens, in the active Name/Builder tab, type a name for the new diagram type in the Name text box (Feeder, for example).
3. Even if the All diagram type created in exercise 3 has no particular characteristics, the new diagram type you are creating will inherit from this All diagram type. In this case, if you decide to create specific behaviors for the parent diagram type, these behaviors will be automatically inherited at the child level. Select All in the Parent Diagram Type drop-down list.
4. Select Custom Query Based Builder from the Schematic Builder drop-down list.

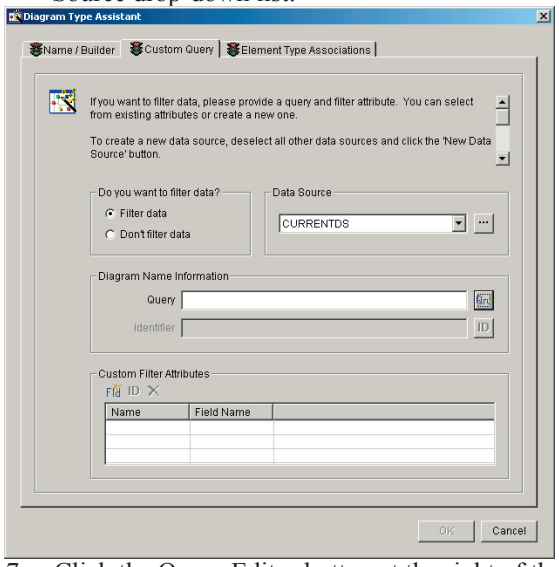


You must now specify the Custom Query parameters for this diagram type.

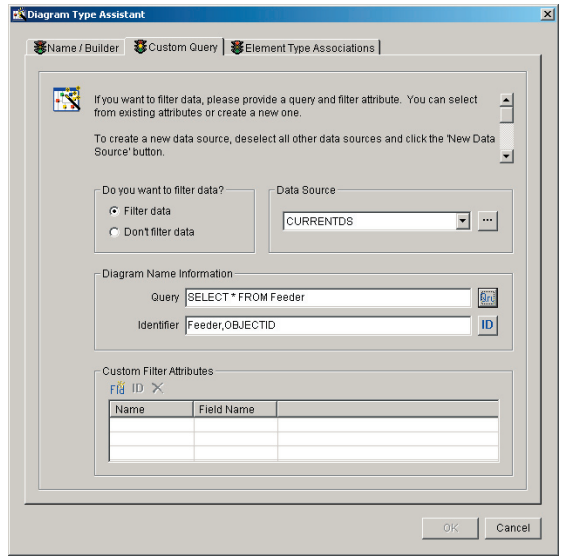
5. Click the Custom Query tab.

Because this new diagram type must filter out the schematic elements that will be associated with it according to the Feeder subnetwork these elements belong to, the query that will return all the schematic diagrams of this type should return the distinct occurrences of the subnetworks stored in the database. In this case, the SQL query for the Feeder diagram type can be defined from the Feeder table that contains all network feeders.

6. Check the Filter data option and click CURRENTDS from the Data Source drop-down list.



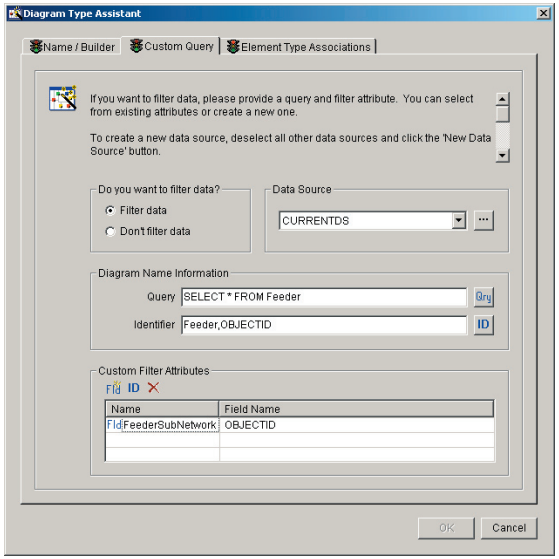
7. Click the Query Editor button at the right of the Query text box.
8. From the Query Editor that opens, click the Feeder table in the Tables list and click All Records so the SELECT * FROM Feeder SQL query automatically displays in the Query area.
9. Click OK to close the Query Editor form.
10. The Identifier Editor automatically opens. In the central text zone type “Feeder” and click the Down arrow to validate this parameter.
11. Then, from the Fields list, click OBJECTID and click the Down arrow to add this field.
12. Click OK to close the Identifier Editor dialog box.



When such a diagram type is created to filter out elements stored in the database, you must define the diagram type attribute that will later be used to filter the schematic elements associated with a given diagram of this type. This way, these schematic elements will be automatically associated with each diagram of this type. In this case, it is the Feeder table’s OBJECTID field that you will need.

13. Click the Create Filter Attribute icon in the Custom Filter Attributes area.
14. Type a name for the attribute by changing the default NewAttribute name into the desired one (FeederSubnetwork in this example), and validate.
15. Then, from the Identifier Editor that automatically opens, click OBJECTID from the Fields list, click the Down arrow, and validate.

All parameters related to your new diagram type are now specified. At the end of this step, the Custom Query tab should display as follows:



You must now specify the elements that will be contained in each feeder diagram implemented by this diagram type. Using the Diagram Type Assistant, those parameters must be set in the Element Type Associations tab.

Creating new schematic element types to filter out nodes and links according to their subnetworks

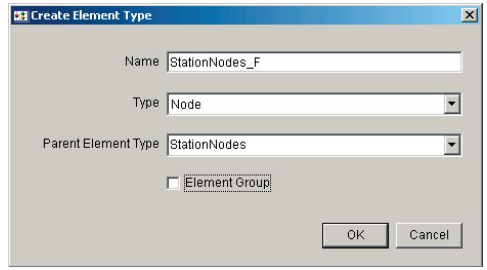
In this section, you will create two new node element types and one new link element type that will be associated with the new Feeder diagram type. The queries that will return the elements of these types will be built according to the FeederSubnetwork attribute (defined for the Feeder diagram type) so they automatically filter out the schematic elements related to one feeder. Using the Diagram Type Assistant, those parameters can be specified through the Element Type Associations tab.

1. Click the Element Type Associations tab in the Diagram Type Assistant.

The Element Type Associations tab should contain the four element types you have created during the previous exercises. Since these element types are not associated with the current diagram type you are specifying, their related checked box at the left is unchecked.
Now you are going to create the new node and link element types that will be associated with your diagram type.

Creating the StationNodes_F new node type

1. Click the Create Element Type button to open the Create Element Type dialog box.
2. In the Name text box, type “StationNodes_F”.
3. From the Type drop-down list click Node.
4. From the Parent Element Type drop-down list, click StationNodes.
5. Uncheck the Element Group check box.



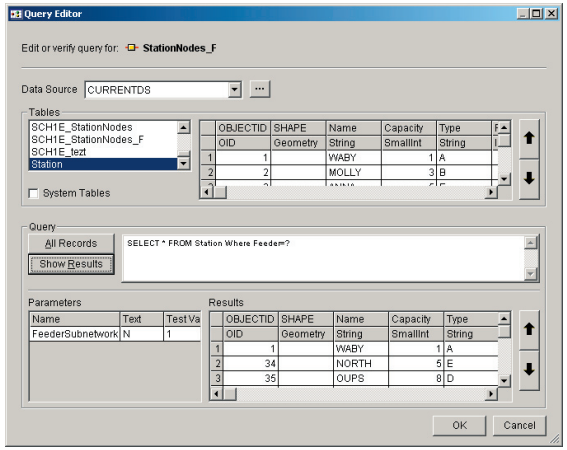
6. Click OK to validate.

The Query Editor automatically opens to allow you to specify the query that will filter out the stations stored in the database according to the FeederSubnetwork attribute values defined for this Feeder diagram type.

The new Associated Element Types tab displays among the Feeder diagram type's tabs. An item corresponding to StationNodes_F should now be displayed in this tab.

7. In the Query zone area, remove the default query and type this one: SELECT * FROM Station Where Feeder=?
8. Then you need to focus on the Parameters area. From the Name drop-down list, click FeederSubnetwork (the attribute you defined for the Feeder diagram type the StationNodes_F is associated with).
9. As this attribute returns numerical values (the OBJECTID field related to this attribute stores numerical values), click N from the Text drop-down list.

If you want to test your query, type "1"," 2", or "3" for the Test Value field and click Show Results (the query result automatically displays in the Results zone).



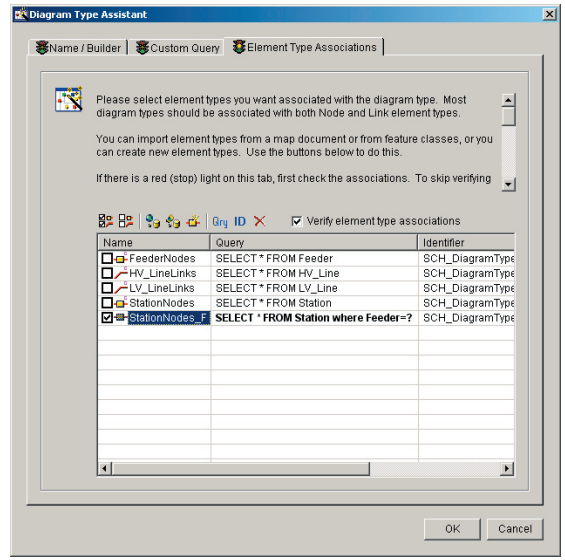
10. Click OK.

The Identifier Editor dialog box automatically opens. By default, as the StationNodes_F element type you are specifying inherits from the StationNodes one, all the fields you specified to build the StationNodes node's identifier are automatically inherited.

11. Check these default inherited fields to ensure the resulting identifiers are unique.
12. Click OK.

The first new StationNodes_F element type definition is finished.

The Element Type Associations tab should now display as follows:



Defining the FeederNodes_F new node type

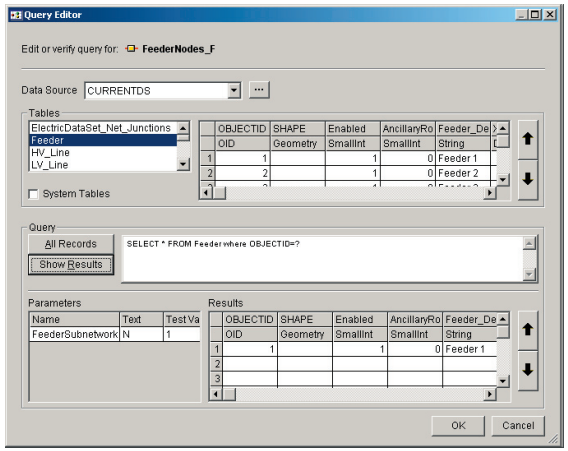
The purpose now is to create the new FeederNodes_F element type that will return the unique feeder related to each Feeder subnetwork diagram. This new node element type will inherit from the FeederNodes element type you created in exercise 3.

1. Click the Create Element Type button to open the Create Element Type dialog box.
2. In the Name field type “FeederNodes_F”, from the Type drop-down list click Node, from the Parent Element Type drop-down list click FeederNodes, uncheck the Element Group check box, and validate.

The Query Editor automatically opens so you can specify the query that will return all the feeder nodes of this type.

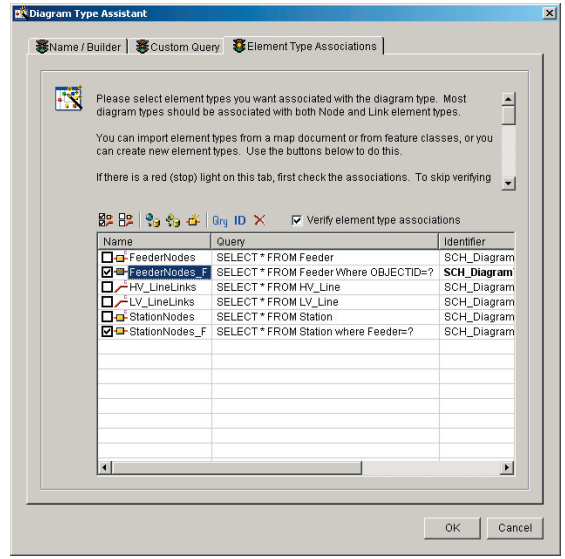
3. In the Query zone area, remove the default query and type this one: SELECT * FROM Feeder Where OBJECTID=?
4. Then focus on the Parameters area. From the Name drop-down list click FeederSubnetwork (the attribute you defined for the Feeder diagram type the FeederNodes_F is associated with).
5. As this attribute returns numerical values (the OBJECTID field related to this attribute stores numerical values), click N from the Text drop-down list.

If you want to test your query, type “1”,“2”, or “3” for the Test Value field and click Show Results (the query result automatically displays in the Results zone).



6. Click OK.
7. The Identifier Editor dialog box automatically opens. Check the default inherited fields to ensure the resulting identifiers are unique and click OK.

At the end of these steps, the FeederNodes_F definition is finished and the Element Type Associations tab should display as follows:



Defining the new LV_LineLinks_F element type

You will now create a new element type link that will be used to filter out all the links stored in the LV_Line table, according to the subnetwork to which they belong. This new link type will inherit from the LV_ LineLinks element type.

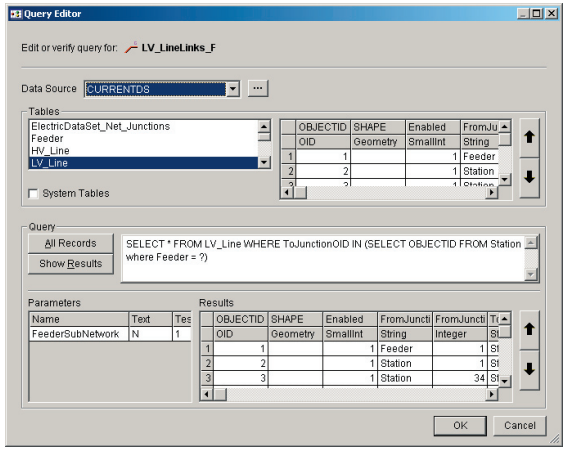
Note that since the HV_Line table contains links that connect Feeder nodes to other Feeder nodes and the new Feeder diagram type will be used to display subnetworks related to a unique feeder, no new link type must be created from the HV_Line table.

1. Click the Create Element Type button to open the Create Element Type dialog box.
2. In the Name field type “LV_LineLinks_F”, from the Type drop-down list click Link, from the Parent Element Type drop-down list click LV_LineLinks, uncheck the Element Group check box, and validate.

The Query Editor automatically opens.

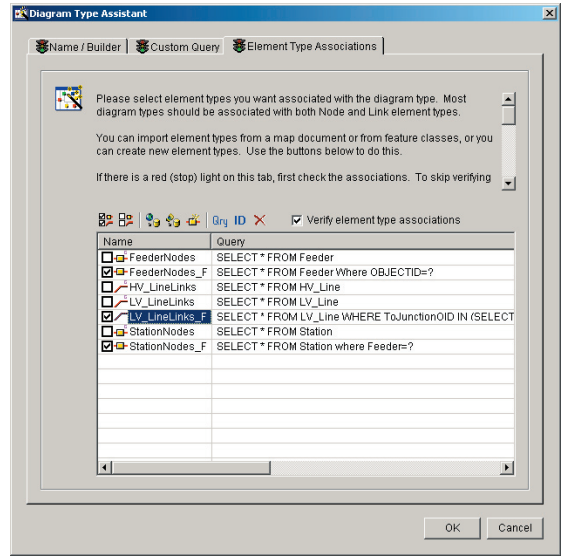
3. Remove the default query set in the Query area and type the following one:
   SELECT * FROM LV_Line WHERE ToJunctionOID IN (SELECT OBJECTID FROM Station where Feeder = ?)
4. In the Parameters area, from the Name drop-down list, click FeederSubNetwork and for the Text parameter select N.

Test the specified query by typing “1”, “2”, or “3” for the Test Value field and click Show Results to verify that the specified query returned the desired records.



5. Click OK.
6. The Identifier Editor dialog box automatically opens. Check the default inherited fields to ensure the resulting identifiers are unique and click OK.

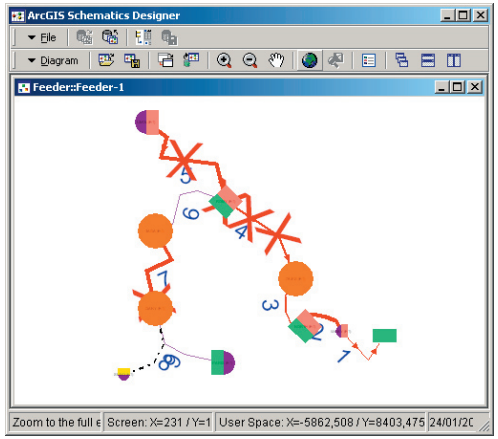
At the end of these steps, the LV_LineLinks_F definition is finished. The Element Type Associations tab should display as follows:



7. Click OK to close the Diagram Type Assistant.
8. From the Designer tree, check that the StationNodes_F, FeederNodes_F, and LV_LineLinks_F element type entries correctly display.
   In particular, look at the LV_LineLinks element type entry. A link can be displayed only when Schematics knows to which origin and extremity nodes the link connects. In this example, because the query defined for this link element type returns all the fields required to compute the inherited OriginNode and ExtremityNode attribute values, each link should correctly connect its origin and extremity nodes for FeederNodes_F and StationNodes_F.
9. Click Save.

You can now check that the new Feeder diagram type correctly generates three diagrams related to each Feeder subnetwork.

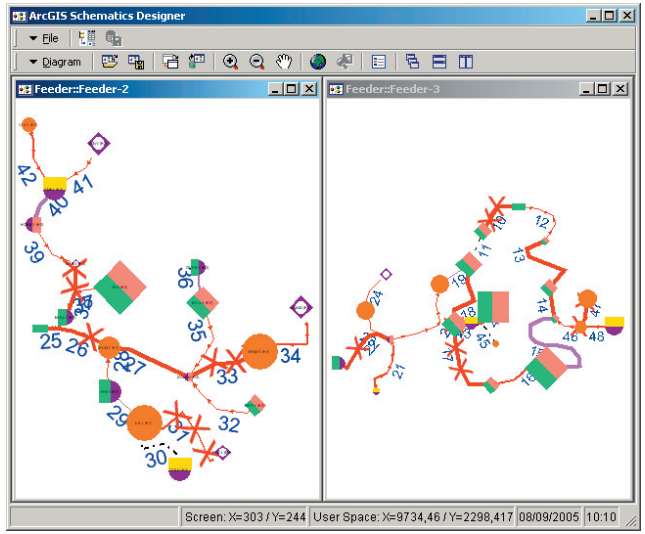
10. Switch to run mode by closing the Designer Editor window.
11. Click Open Diagrams.
12. From the Select Diagram To Open dialog box that opens, click Feeder in the Diagram Type drop-down list.
13. The Diagram Name drop-down list should contain three items (Feeder-1, Feeder-2, and Feeder-3), one for each feeder stored in the Schematics_Designer_Tutorial database. Click Feeder-1.
14. Click OK. The Feeder-1 schematic diagram should open as follows:



Note that the diagram names available from the Diagram Name drop-down list of the Select Diagram To Open dialog box are those returned by the Identifier diagram type parameter.

Note also that because the new StationNodes_F, FeederNodes_F, and LV_LineLinks_F element types inherit from the StationNodes, FeederNodes, and LV_LineLinks element types, all properties defined for the parent element types are automatically reported on the children element types. As the queries set for the new element types return the same parent fields needed by these inherited properties, the properties' representation is effective.

The following screen shots show how the Feeder-2 and Feeder-3 diagrams should display:



15. Close the Feeder-2 and Feeder-3 diagrams.
16. Click Edit to switch to Design mode.

Redefining inherited properties for the schematic elements contained in the Feeder diagrams

Due to the inheritance, the Label1 and Label2 properties defined for the StationNodes and LV_LineLinks element types are automatically reported on the StationNodes_F and LV_LineLinks_F element types. However, inherited properties can also be redefined at the child level if needed.

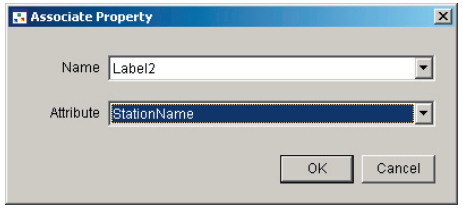
In this section, you will redefine these property labels as follows:

- In Feeder subnetwork diagrams, the Label2 textual property will display each station NAME field value stored in the database (instead of displaying the inherited, formatted StationLabel attribute value).
- In Feeder subnetwork diagrams, the Label1 textual property will display a new label that will concatenate its origin node OBJECTID and extremity node OBJECTID as follows: FromJunctionOID-ToJunctionOID.

Associating an inherited property with an inherited attribute

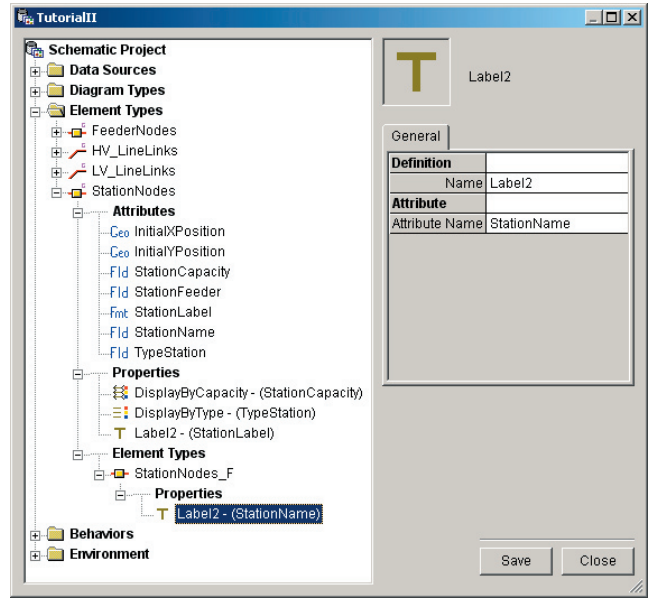
The StationName attribute field defined for the StationNodes element type corresponds to the NAME field stored in the database.
As the StationNodes_F element type inherits from the StationNodes element type, this attribute is automatically inherited at the child level. At the StationNodes element type level, the Label2 property is associated with the formatted StationLabel attribute. In the Feeder diagrams, you want to associate the Label2 inherited property with the StationName inherited attribute. For the StationNodes_F element type, the Label2 property redefining consists of reassociating this inherited property with the inherited StationName attribute.

1. Right-click the StationNodes_F entry and click Associate Property.
2. From the Associate Property dialog box that opens, from the Name drop-down list, click the inherited Label2 property.
3. Then, from the Attribute drop-down list, click the StationName attribute among all the inherited attributes available.



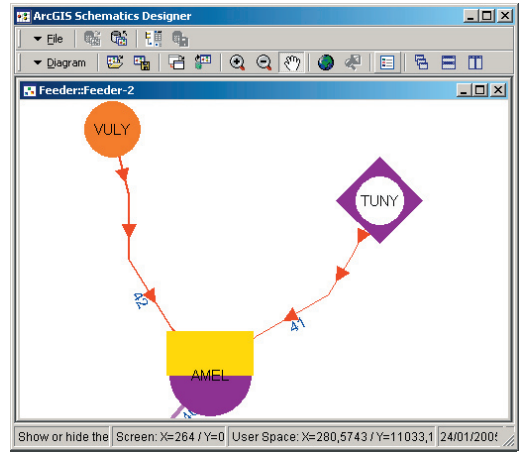
4. Click OK to validate.

The property entry should display as follows:



5. Click Save.

If you want to test the inherited property label redefinition, switch to Run mode and open the Feeder-1, Feeder-2, or Feeder-3 diagram. After zooming in on a set of stations, the label should display in a similar way as in the following screen shot:



Defining a new attribute at the child element type level and associating an inherited property with that attribute

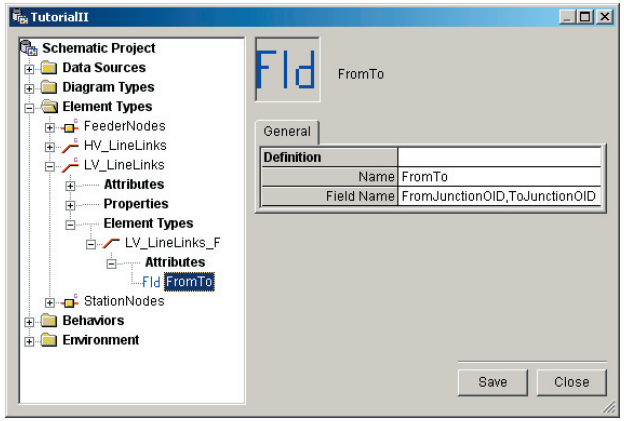
At the LV_LineLinks element type level, the Label1 property is associated with the LVLine_OID field attribute. For the Feeder diagrams, you will have to change the inherited property label so that this property displays a formatted label built as follows: FromJunctionOID-ToJunctionOID.

You must start by creating a new attribute at the LVLineLinks_F level to build this new label. Since a field attribute can concatenate several fields returned by its related element type's query, with each concatenated field value being hyphenated, this label will be defined as a field attribute.

1. Right-click the LVLineLinks_F entry and click Create Attribute.
2. From the Create Element Type Attribute dialog box that opens, in the Name field type a name (FromTo for example), from the Type drop-down list click Field Attribute, and validate.
3. The Identifier Editor dialog box automatically opens. Double-click the FromJunctionOID field available from the Fields list so this field is added to the Identifier list,
4. Repeat this operation for the ToJunctionOID field.
5. Click OK to validate.

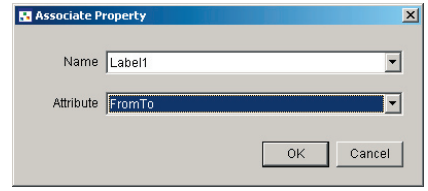
The Identifier Editor and Create Element Type Attribute dialog boxes close.

The FromTo attribute's General tab should display as follows:



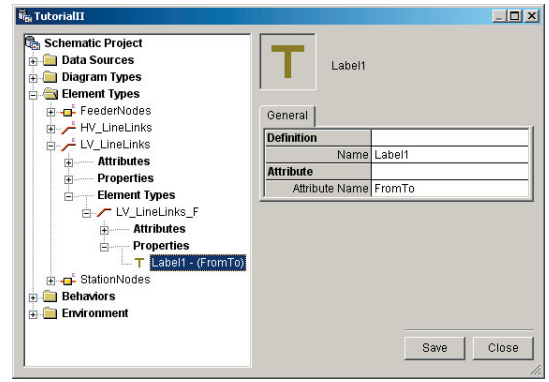
Now, the Label1 property redefinition only needs the reassociation of this inherited property with the new FromTo attribute defined at the LVLineLinks_F level.

6. Right-click the LVLineLinks_F entry and click Associate Property to open the Associate Property dialog box.
7. From the Name drop-down list, click the inherited Label1 property and from the Attribute drop-down list, click the FromTo attribute among all the attributes.



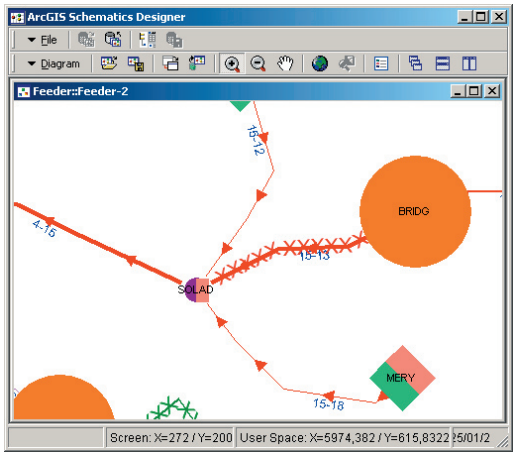
8. Click OK to validate.

The newly associated property's General tab should display as follows:



9. Click Save.

If you want to test the inherited property label redefinition, switch to Run mode and open the Feeder-1, Feeder-2, or Feeder-3 diagram. After zooming in on a set of links, the label should display in a similar way as in the following screen shot:



The purpose of this exercise is to create and specify the parameters required to manage associations between features/objects and schematic elements so the Propagate Schematic Selection To Map and Propagate Map Selection To Schematic commands work when you open the All, Feeder-1, Feeder-2, or Feeder-3 diagram in ArcMap, and so the Identify Results dialog box displays properties related to the clicked schematic element and those related to its associated feature(s)/object(s) when you are using the Identify tool on any schematic element contained in these diagrams.

For diagrams that work with the Custom Query Based Builder, Schematics automatically tries to manage the associations between features/objects contained in a geodatabase and schematic elements contained in these diagrams by searching for three predefined user data values on the schematic elements—FOID, FCN, and FDSN:

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

Note that only the FOID user data is mandatory for Schematics to manage these associations. If the FCN user data is missing for the schematic element, Schematics searches for a feature class or table whose name corresponds to the name of the element type that implements the schematic element. If the FDSN user data is missing, Schematics uses the data source specified for the element type query.
The FOID, FCN and FDSN user data value can return unique values or values list when a schematic element is associated with several features/objects that can be stored in a unique or several feature class(es)/object table(s) coming from a unique datasource or different datasources. To associate schematic elements with a set of features/objects, the user data are associated with formatted attributes with each feature/object value separated by the semicolon (;) character in a list.

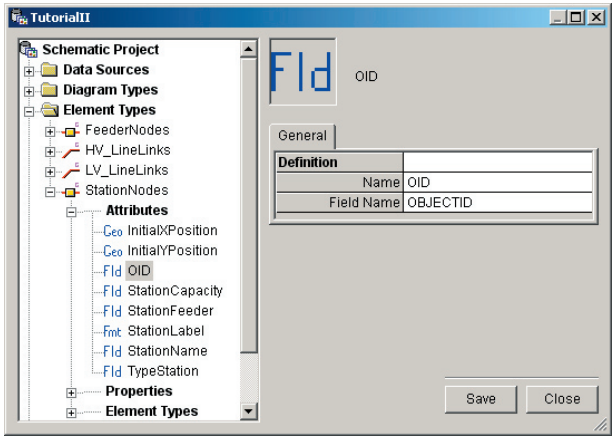
Creating the user data for the StationNodes element type

User data is always related to an attribute, and it is this attribute that returns the user data values. Any user data definition begins with an attribute creation that will return the user data value.

The schematic elements implemented by the StationNodes element type must be associated with the features contained in the Station feature class stored in the Schematics_Designer_Tutorial database. Since the CURRENTDS data source used to query the schematic elements of this type connects the Schematics_Designer_Tutorial database, the FDSN user data doesn't need to be specified. You will only specify the FOID and FCN user data values.

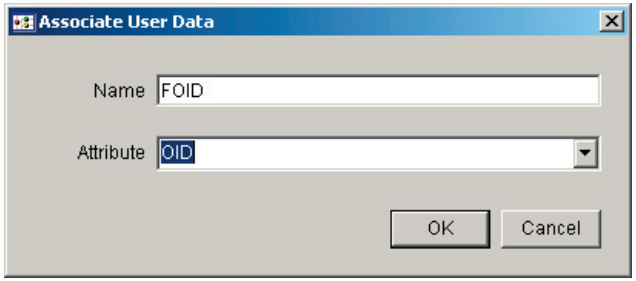
1. Right-click StationNodes from the Designer tree and point to Create Attribute.
2. From the Create Element Type Attribute dialog box, type a name for the new attribute (OID for example).
3. Click Field Attribute for the Type and validate.
4. Then from the Identifier Editor dialog box that opens, double-click OBJECTID from the Fields list and validate.

At the end of these steps, the OID attribute entry should display as follows:

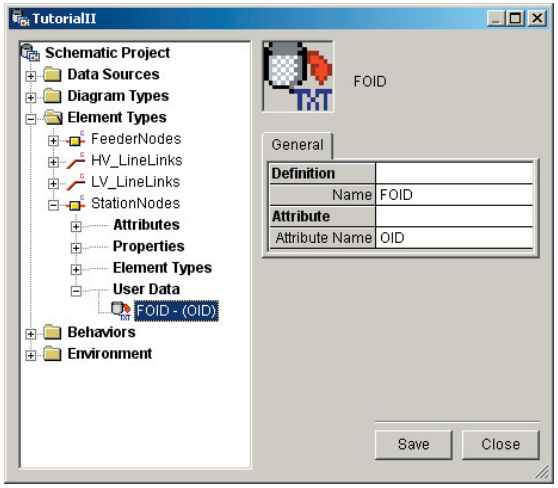


You must now create the FOID predefined user data related to this attribute.

5. Right-click StationNodes from the Designer tree and point to Associate User Data.
6. From the Associate User Data dialog box that opens, type “FOID” in the Name text box.
7. Click the attribute you have just created from the Attribute drop-down list (OID in this example).

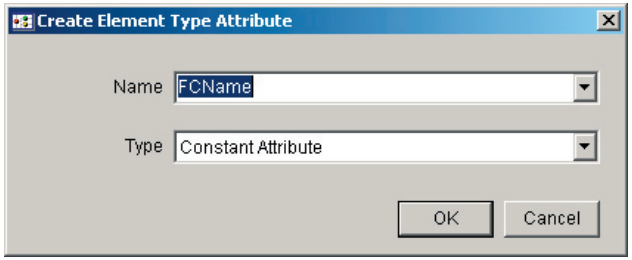


8. Click OK.



The FOID user data definition for the StationNodes elements is now finished.
You must now create the FCN user data that will specify the feature class name of the StationNodes elements. Since the StationNodes elements represent the features stored in the Station feature class table, creating a constant attribute whose value is “Station” is the solution here.

9. Right-click StationNodes from the Designer tree and click Create Attribute.
10. Then from the Create Element Type Attribute dialog box, type a name for the new attribute (FCName, for example).
11. Click Constant Attribute from the Type drop down list.



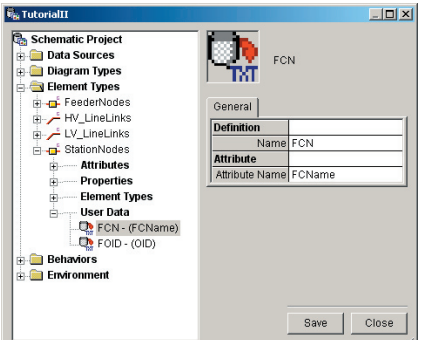
12. Click OK.

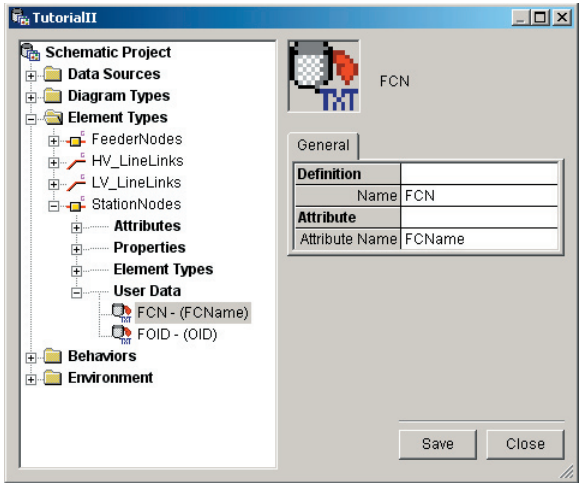
A new entry corresponding to this attribute should now display in the Designer tree. Now you need to complete this attribute definition.

13. Type “Station” in the Value text box and validate.

Now you need to create the FCN user data you want to be associated with this attribute.

14. Right-click StationNodes from the Designer tree and point to Associate User Data.
15. In the Associate User Data dialog box, type “FCN” for the user data Name.
16. Click the attribute name you have just created (FCName, in this example) and validate.

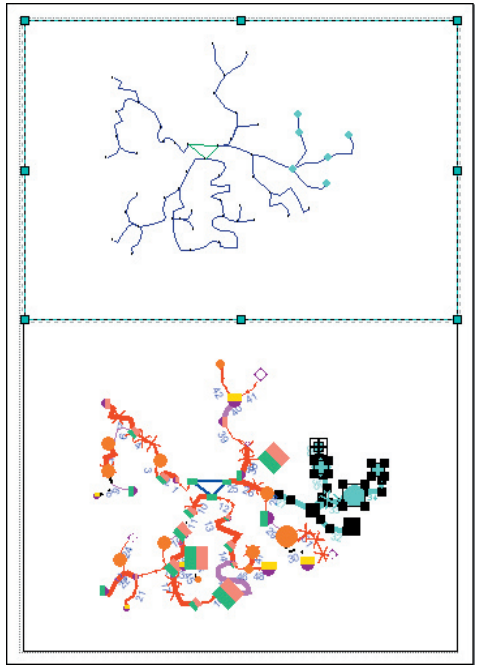




You can now test the impact of these definitions in ArcMap.

17. Click Save and close ArcGIS Schematics Designer.
18. Start ArcMap with a new, empty map.
19. Click Add Data, browse to the Schematic_Designer_Tutorial database that should have been installed in C:\arcgis\ArcTutor\Schematics\Designer, click ElectricDataSet, and click Add.
20. Click Open Schematic Diagrams.
21. From the Select schematic diagrams to open dialog box that opens, browse to the TutorialII schematic dataset contained in the Schematic_Designer_Tutorial database and select All.
22. Click Add.
23. Then click View, point to Layout View, and arrange the geographic and schematic data frames.
24. Click the Select Schematic Elements tool and select any part of the network in the All diagram that contains StationNodes nodes.
25. Click Propagate Schematic Selection To Map.

The Station features related to the StationNodes schematic nodes currently selected in the All diagram should be selected in your map.



26. Clear the selection features set from your map.
27. Use the Select Features tool to create a new selection set of features in your map.
28. Click Propagate Map Selection To Schematic.

The StationNodes nodes associated with the Station features selected in your map automatically select in the All schematic diagram.
Note that the Propagate Schematic Selection To Map and Propagate Map Selection To Schematic commands work only for the schematic nodes and feature points contained in the Station feature class. For the other elements and features to also be automatically selected during the commands execution, you need to create the appropriate user data on the FeederNodes, LV_LineLinks and HV_LineLinks element types.

29. Save your .mxd file and close ArcMap.
30. Edit the TutorialII schematic project in ArcGIS Schematics Designer.

Creating the user data for the FeederNodes, HV_ LineLinks, and LV_LineLinks element types

1. Repeat the previous steps 1 to 8 for each FeederNodes, HV_ LineLinks, and LV_LineLinks element type to create the FOID user data. This user data must be associated with an attribute field link to the OBJECTID field returned by each element type query.

Please note that for the LV_LineLinks element type, an attribute that returns the OBJECTID field values should already exist (In this example, it is called LV_LineOID). So, in this case, to create the predefined FOID user data for LV_LineLinks, you only need to associate the FOID user data with the LV_LineOID attribute.

2. Repeat the previous steps 9 to 16 for the FeederNodes, HV_LineLinks and LV_LineLinks element types to create the FCN user data.This user data will be associated with an attribute constant whose value is Feeder, HV_Line, or LV_Line depending on the element type.
3. Click Save and Exit Designer.
4. Load your .mxd file in ArcMap and test the Propagate Schematic Selection To Map and Propagate Map Selection To Schematic commands again.
   For any schematic element selected in the All diagram, Propagate Schematic Selection To Map should now automatically select the associated feature in your map.
   Conversely, for any feature selected in your map, using the Propagate Map Selection To Schematic should select the associated feature in the All diagram.

Since any element type associated with the diagrams implemented by the Feeder diagram type inherits from the StationNodes, FeederNodes, or LV_LineLinks element types, the Propagate Map Selection To Schematic and Propagate Schematic Selection To Map commands should also work on the Feeder diagrams.

Note: In this example, as each schematic element is associated with a unique feature, the FOID, FCN and FDSN user data return a unique value. For an example of schematic elements associated with several features/objects, you can edit the ElecDemo schematic dataset in the ElecDemo personal geodatabase within Designer—the ElecDemo database should have been installed in the ArcTutor\Schematics\ Schematics folder. Let's have a look to the InsidePlant_RootNodes and InsidePlant_RootLinks father element type's definitions. In this case, each schematic element which inherits from these element types is both associated with a feature stored in the ElectricNetwork feature dataset and with an object stored in the Inside_Nodes or Inside_Links object tables. Formatted attributes associated with the predefined FOID and FCN user data have been created to specify these multiple associations.

Note: Instead of creating the FCN user data on each parent element type, it would be easier to specify the Associated Feature Class/Table field value for each parent and child element type in the General tab—that is, for StationNodes, FeederNodes, LV_LineLinks, StationNodes_F, FeederNodes_F, and LV_LineLinks_F element type.
From each General tab related to these element types, you only needed to:

1. Click the Ellipse button that displays when you click the Associated Feature Class/Table field.
2. From the Associated Feature Class/Table dialog box that opens, specify the datasource (Data Source field) and feature class or table name (Name field) you want to be associated with this element type.

In this case, only the FOID user data definition will be required.

Exercise 7: Relating schematic elements contained in schematic diagrams

The purpose of this exercise is to learn about the parameters that can be specified to relate schematic elements contained in a schematic diagram to each other. Schematic elements can be related to each other by specifying rules or executing custom code during the diagram loading. For example, schematic nodes can be displayed as containers around schematic elements by specifying a Spatial Query rule or Relationship rule. Schematics can also automatically display nodes or drawings as containers around schematic element groups when two predefined user data values are specified for the elements of those groups—PEN and PTN:

- The PEN user data value must specify the name of the container that will be automatically related to the element.
- The PTN user data value indicates the name of the schematic element type that implements this container.

Note that only the PEN user data is mandatory for Schematics to manage the container relations. The PTN user data is required only when several schematic elements contained in the schematic diagram have the same identifier.

In this exercise, you will have to customize the All diagram display so each FeederNodes nodes and their related StationNodes nodes display in a specific container, one container per feeder. The OBJECTID field available from the Feeder feature class and the Feeder field stored in the Station feature class will be used to identify the container that will be related to each of them. You must start by creating a new element type to implement the containers you want to be displayed in your All diagram.

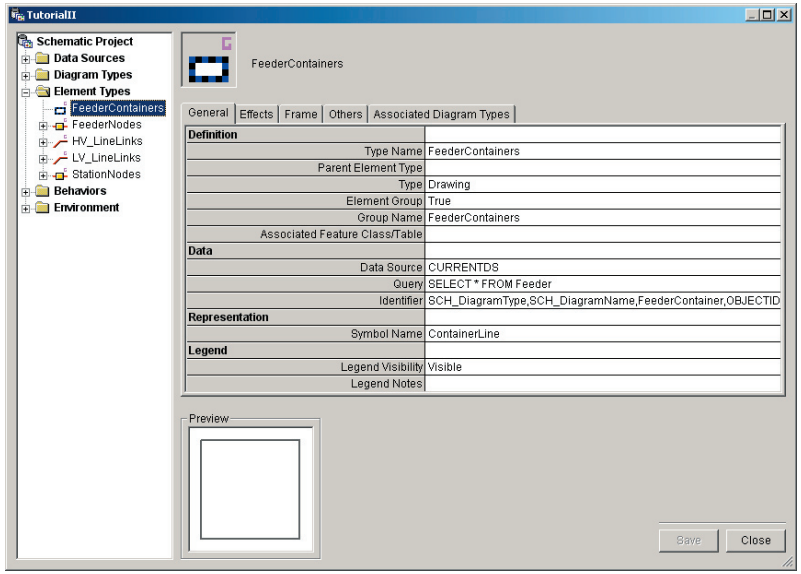
Creating the new FeederContainers drawing element type

1. Edit your schematic project within Designer.
2. Right-click Element Types in the Designer tree and click Create.
   The Create Element Type dialog box opens.
3. Type FeederContainers in the Name field.
4. Click Drawing from the Type drop-down list.
5. Keep the Element Group option checked by default and validate.
   The new drawing element type subentry is automatically created below the Element Types main entry in the Designer tree.
6. Click the General tab and click the CURRENTDS data source from the Data Source drop-down list.
7. Click the Query field and click the Ellipsis button that displays.
8. From the Query Editor dialog box that opens, double-click the Feeder table where all records corresponding to the containers you want to implement are stored.
9. Click OK to validate.
10. Then, click the General tab's Identifier field and click the Ellipsis button that displays to open the Identifier Editor dialog box.

As you had previously done for your element type identifier, each FeederContainers drawing will be identified by concatenating the diagram type name and the diagram name where it will be displayed, the FeederContainer string, and the OBJECTID field values.

11. Double-click the SCH_DiagramType and SCH_DiagramName items available at the end of the Fields list so these two fields appear in the Identifier list.
12. Type "FeederContainer" in the central text zone and validate by clicking the Down arrow.
13. Double-click the OBJECTID field from the Fields list to add this field to the Identifier list.
14. Click OK.
    The Identifier Editor dialog box closes.
15. Then click the Symbol Name field and click the Select Symbol button.
16. From the Symbol Browser that opens, click the ContainerLine symbol and click OK.

At the end of these steps, the General tab related to your new FeederContainers element type should display as follows:



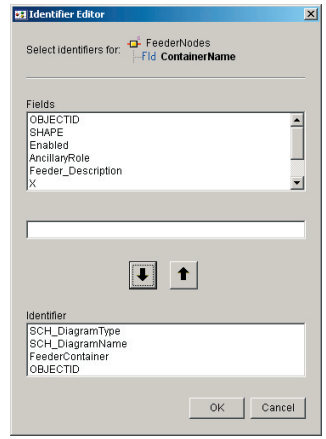
You must now associate this new element type with the All diagram type:

17. Right-click the FeederContainers element type entry in the Designer tree and click Edit Associations.
18. From the Diagram Type Associations dialog box that opens, check All and validate.
19. Save your schematic project definition.

Creating the user data for the FeederNodes element type

The purpose now is to create the PEN user data that will specify the FeederContainer identifier for each FeederNodes node you want to be related to a container. You must build an attribute that will be associated with the PEN user data.

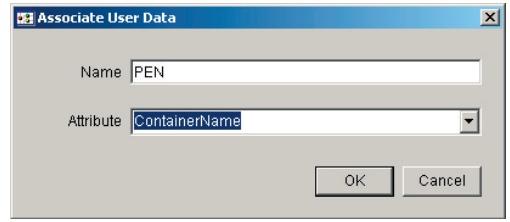
1. Right-click the FeederNodes element type entry in the Designer tree and click Create Attribute.
2. In the Create Element Type Attribute dialog box, type ContainerName in the Name field.
3. Keep Field Attribute for the Type field and validate.
4. Then, from the Identifier Editor, double-click the SCH_DiagramType and SCH_DiagramName items at the end of the Fields list so these two items appear in the Identifier list.
5. Type “FeederContainer” in the central text zone and validate this string by clicking the Down arrow.
6. Double-click the OBJECTID field from the Fields list to select it.



7. Click OK to validate.

You must now associate this newly created attribute with the PEN userdata.

8. Right-click the User Data entry that displays under the FeederNodes entry in the Designer tree and click Associate.
9. Type PEN for the Name field in the Associate User Data dialog box that opens.
10. Click ContainerName in the Attribute drop-down list.



11. Click OK.

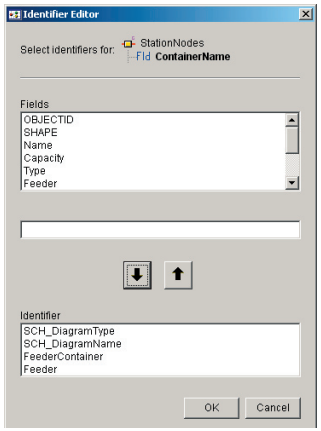
Creating the user data for the StationNodes element type

You must now operate in the same way to create the same PEN user data for the StationNodes nodes.

1. Right-click the StationNodes element type entry and click Create Attribute.
2. In the Create Element Type Attribute dialog box, type ContainerName in the Name field, keep Field Attribute for the Type field and validate.

In the Station table, this is the Feeder field that provide the OBJECTID of the Feeder related to each station.

3. So, from the Identifier Editor, double-click SCH_DiagramType and SCH_DiagramName, type “FeederContainer” in the central text zone and validate this string by clicking the Down arrow.
4. Then double-click the Feeder field from the Fields list to select it.

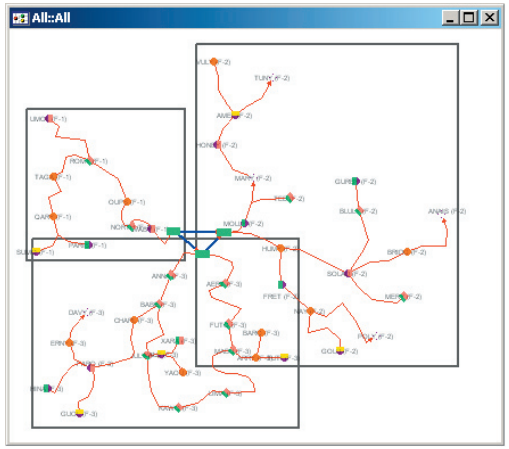


5. Click OK to validate.

Now you need to associate this ContainerName attribute with the PEN userdata as you previously did for the FeederNodes.

6. Right-click the User Data entry that displays under the StationNodes entry in the Designer tree and click Associate.
7. Type PEN for the Name field in the Associate User Data dialog box that opens.
8. Click ContainerName in the Attribute drop-down list and validate.
9. Save your schematic project definition.

The PEN user data definition for the StationNodes and FeederNodes element types is now complete. The definition of the second predefined PTN user data required when a same identifer is used for several schematic elements contained in a single diagram is not necessary in this case because each schematic element identifier is unique. If you switch to Run mode and open the All diagram, it should display as follows:



Exercise 8: Defining schematic behaviors

In this exercise, you will learn how to define schematic behaviors. A behavior defines how an application responds to a user's actions. Behaviors can be applied to Schematics diagrams, elements, and views. Note that when schematic diagrams open in the Schematics Designer environment, behaviors can also be applied to legend entries. But when these diagrams display in the ArcMap environment as schematic layers, behaviors defined for legend entries cannot be intercepted.

Defining behaviors that will impact schematic diagrams

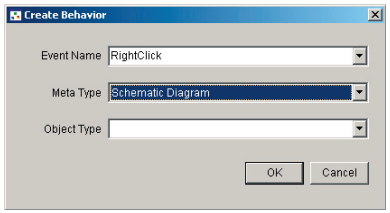
In this section, you will define a behavior impacting any schematic diagram (or view) when it is right-clicked. Generally, the main commands needed from a diagram are as follows:

- Fitting all the elements in the diagram
- Repositioning the elements displayed in the diagram according to their initial positions

You are going to create a behavior that will display a context menu for launching these three commands:

To create behaviors that impact diagrams:

1. Right-click the Behaviors entry in the Designer tree and click Create.
2. From the Create Behavior dialog box that opens, choose RightClick from the Event Name drop-down list and click Schematic Diagram from the Meta Type drop-down list.
   In this example, because the behavior you are defining will impact any diagram defined for the schematic project, keep the Object Type parameter empty.

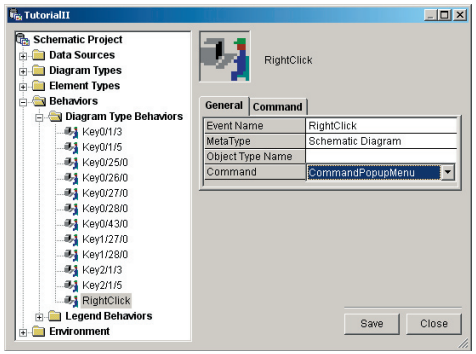


3. Click OK to validate.

The new RightClick behavior entry is automatically created under the Diagram Types Behaviors main entry.

You must now specify the command that will be automatically triggered each time a schematic diagram is right-clicked. The predefined command used to display a context menu is CommandPopupMenu:

4. From the new behavior's General tab, click CommandPopupMenu from the Command drop-down list.

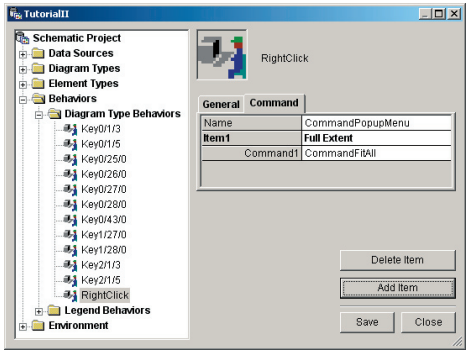


The purpose now is to define the CommandPopupMenu parameters. The Command tab related to a behavior definition always lists the parameters for the main command set in the General tab's Command parameter.

5. Click this Command tab.

In this example, the first menu item will be used to fit all the elements contained in the diagram.

6. Type a label for the first item of the context menu in the Item1 parameter field (Full Extent, in this example) and press Enter.
7. Then, from the Command1 drop-down list, click CommandFitAll.



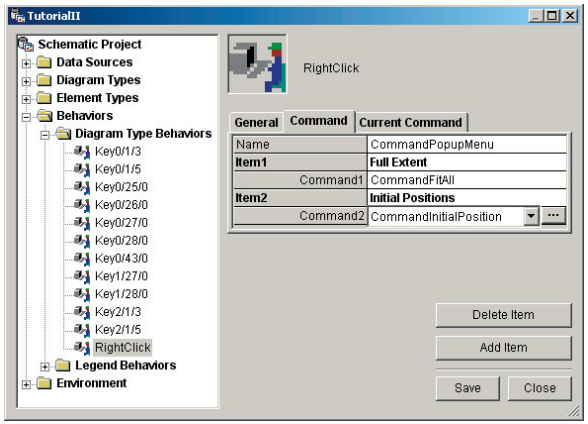
The first item definition is finished.

You must now define the second menu item that will enable the Pan mode:

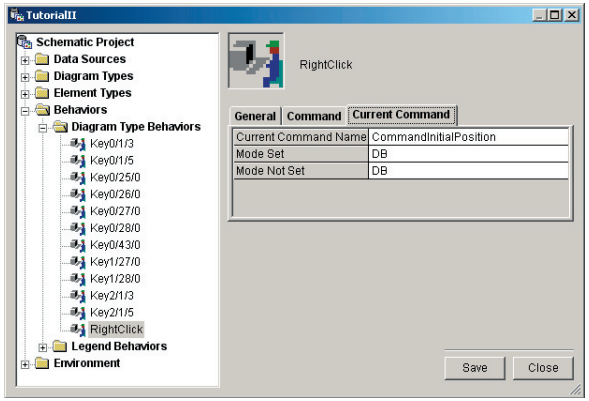
8. Click Add Item to create the second context menu item.
9. Type the item label (Initial Positions, in this example).
10. Then, click CommandInitialPosition from the Command2 drop-down list and press Enter.

The first predefined command you have used for the previous item menu definition was a simple command without parameters. In this case, the CommandInitialPosition is a parameterized command.

11. Click the Command2 parameter field. An Ellipsis button should display at the end of the command line.



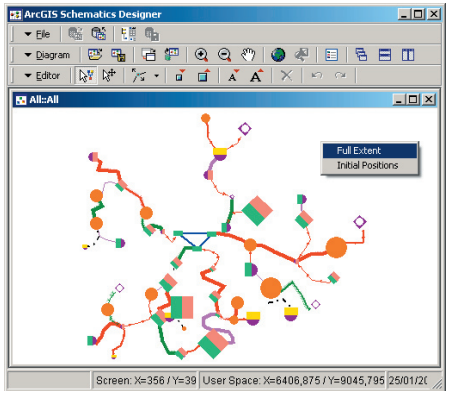
12. Click this Ellipsis button to display the Current Command tab where the parameters related to the CommandInitialPositions are displayed.
13. The Current Command tab content should appear as follows:



The default DB value set for the ModeSet and ModeNotSet parameters is what you need to reposition the elements in the diagram at their initial coordinates.
For more details on the CommandInitialPosition command parameters, open the Schematics Designer Help system.

14. Click Save to save the new specified parameters.

If you want to test this context menu display, switch to Run mode, open a diagram, and right-click this diagram background. The context menu should display as follows:



Defining behaviors that will impact schematic elements

In this section, you will create behaviors that will impact the StationNodes and LV_LineLinks element types when an element of this type is right-clicked.

For example, the context menu displayed from any right-clicked StationNodes station will allow you to:

- Rotate all stations by 45 degrees.
- Center the clicked node in the view.
- Activate or deactivate the Label2 property label display.
- Activate or deactivate the DisplayByCapacity property graphic effects.
- Activate or deactivate the DisplayByType property graphic effects.

From any right-clicked LV_LineLinks link, the context menu that will be displayed will allow you to:

- Activate or deactivate the Label1 property label display.
- Activate or deactivate the DisplayByCategory property graphic effects.
- Activate or deactivate the DisplayByRate property graphic effects.

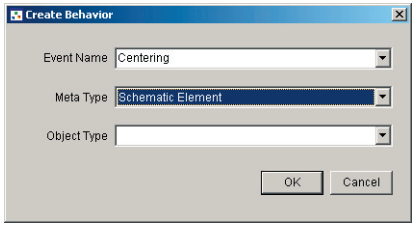
Note: Since StationNodes_F and LV_LineLinks_F element types inherit from StationNodes and LV_LineLinks element types, behaviors defined for StationNodes and LV_LineLinks element types will also impact StationNodes_F and LV_LineLinks_F element types.

Defining the sample behavior related to the StationNodes type

If the first, third, fourth, and fifth context menu items you have to define for the StationNodes element type right-clicked node will correspond to a single parameterized command, the second one that will automatically center the clicked node in the view must chain several commands. In fact, the predefined command used to center an element in a diagram (CommandCenterObject) works on a selected element only. Since right-clicking a node doesn't select it, you will have to select the node (using the CommandSelectObject predefined command) before centering it. But the predefined CommandContainer command usually used to chain several commands cannot be called directly from a context menu item. In this case, the solution consists of defining a user event to trigger the command chaining that you will call from the context menu item through a CommandDelegate command.

The following steps describe this user event behavior:

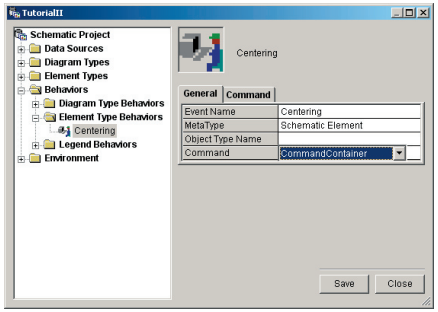
1. Right-click the Behaviors entry in the Designer tree and click Create.
2. From the Create Behavior dialog box that opens, type a name (Centering, for example) for the Event Name parameter that will be used to reference this behavior.
3. Click Schematic Element from the Meta Type drop-down list.



4. Click OK to validate.

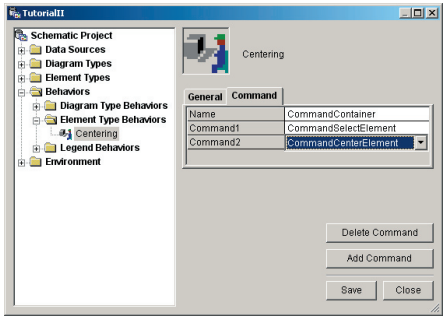
A new Element Type Behaviors entry should display under the Behaviors main entry in the Designer tree. The newly created Centering behavior should display under Element Type Behaviors.

5. From the new behavior General tab, click CommandContainer from the Command drop-down list.



6. Click the Command tab where the CommandContainer parameters must be specified.
7. From the Command1 drop-down list, click CommandSelectElement, the first predefined command that will be used to select the clicked element.
8. Click Add Command to create the second command that will chain with the first.

9. From the Command2 drop-down list, click CommandCenterElement, the predefined command that will then be used to center the clicked element.

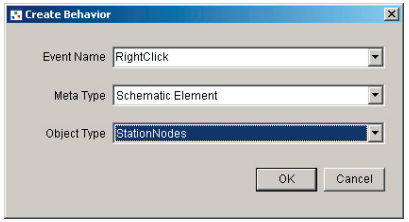


10. Click Save.

Note: This user event definition can end by calling the CommandUnselectAll command so that element is automatically deselected after it is centered in the diagram.

Because this user event is now defined, each item menu you want to be displayed when a StationNodes node is right-clicked can be easily defined and customized as follows:

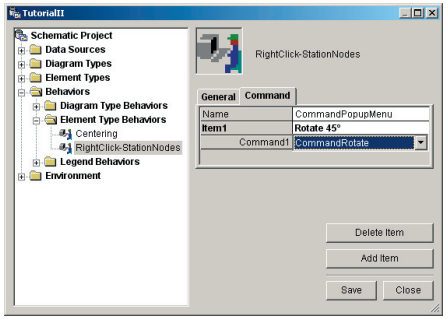
11. Right-click the Behaviors entry in the Designer tree and click Create.
12. From the Create Behavior dialog box, choose RightClick from the Event Name drop-down list, click Schematic Element from the Meta Type drop-down list, and click StationNodes from the Object Type drop-down list.



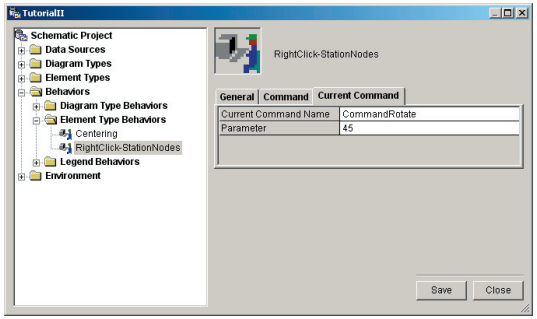
13. Click OK.
14. From the new behavior General tab, click CommandPopupMenu from the Command drop-down list.
15. Then, click the Command tab where parameters related to the CommandPopupMenu must be specified.

Start defining the command that will correspond to the first context menu item. Clicking the first menu item must rotate all stations by 45 degrees:

16. Type the first item that will appear in the menu (Rotate 45°, in this example).
17. Click CommandRotate from the Command1 drop-down list.



18. Click the Ellipsis button that automatically appears on the right of the Command1 field to display its associated Current Command tab where the CommandRotate command parameters are available.
19. Click this Current Command tab and specify the Parameter value that corresponds to the angle that will be used to rotate the nodes (45, in this example).



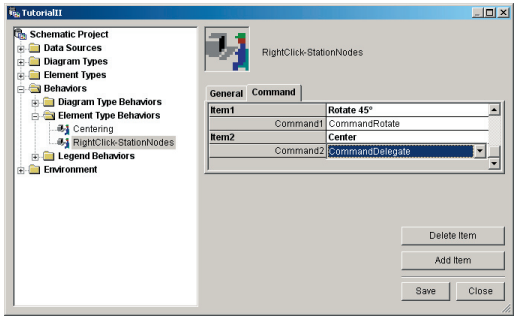
Save the specified parameters.

You must now define the second context menu item. This new menu item will be used to center the clicked node in the diagram:

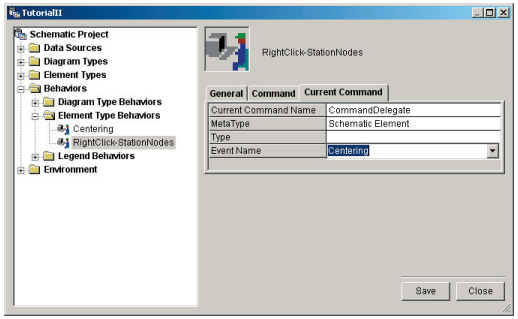
20. Click the Command tab and click Add Item.
21. Type the second item that will appear in the context menu (Center, in this example).

The command you must specify here is not a simple, predefined command. You must launch the execution of the Centering behavior you previously defined. The CommandDelegate is the predefined command you must use to trigger the execution of an existing behavior:

22. Click CommandDelegate from the Command2 drop-down list.



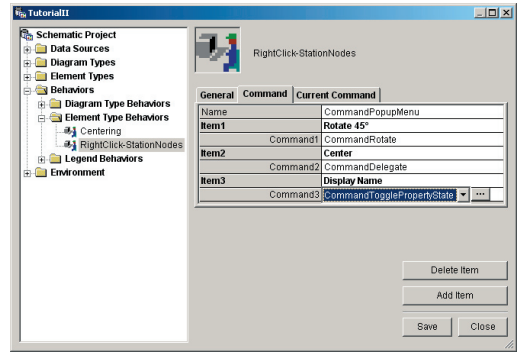
23. Click the Ellipsis button on the right of the Command2 field to display its associated Current Command tab and click this tab.
24. The Meta Type parameter default value is the correct one. Click the Centering behavior from the Event Name drop-down list.



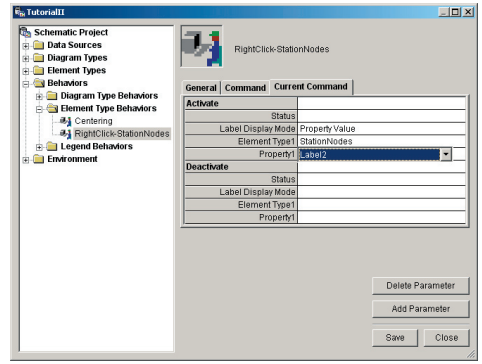
Save the specified parameters.

You must now define the third context menu item. Clicking this new menu item must activate or deactivate the Label2 property label display.

25. Click the Command tab and click Add Item to set the third context menu item.
26. Type the third item that will appear in the context menu (Display Name, in this example).
27. Select CommandTogglePropertyState from the Command3 drop-down list.



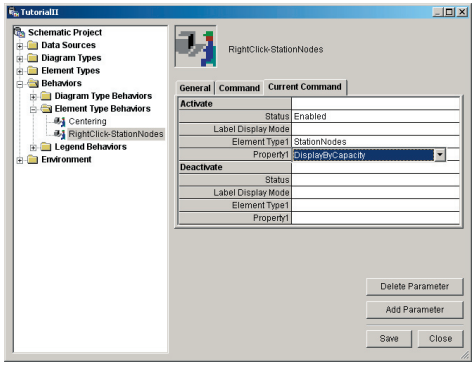
28. Click the Ellipsis button on the right of the Command3 field to display its associated Current Command tab and click this tab.
29. Here, as you want to enable/disable the Label2 property label display, select Property Value from the Label Display Mode drop-down list so that the property label display is turned on/off, select StationNodes from the Element Type1, and select Label2 for the Property1 drop-down list.



Save the specified parameters.

The purpose now is to define the fourth context menu item that will be used to activate or deactivate the DisplayByCapacity property's graphic effects.

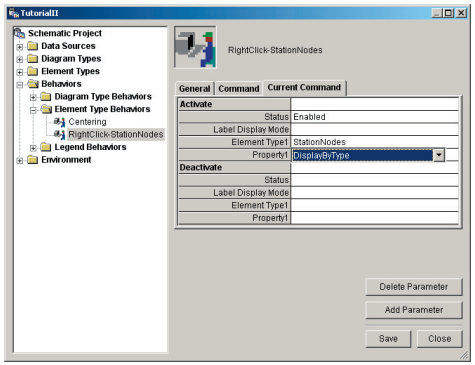
30. Click the Command tab and click Add Item to set the fourth context menu item.
31. Type the fourth item label that will appear in the context menu (Display by Capacity, in this example).
32. Select CommandTogglePropertyState from the Command4 drop-down list.
33. Click the Ellipsis button on the right of the Command4 field to display its associated Current Command tab and click this tab.
34. Here, since you want to enable or disable the DisplayByCapacity property's graphic effects, click Enabled from the Status drop-down list so the property graphic effects are turned on or off according to the current property status. Click StationNodes in the Element Type1 field and click the related property, DisplayByCapacity, for the Property1 field.



Save the specified parameters.

It only allows you to define the last context menu item that will be used to activate or deactivate the DisplayByType property's graphic effects.

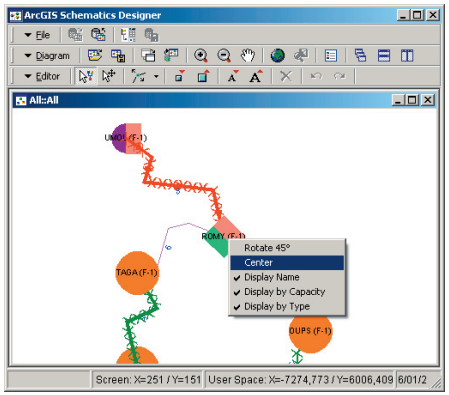
35. Click the Command tab and click Add Item to set the fifth context menu item.
36. Type the item that will appear in the context menu (Display by Type, in this example).
37. Click CommandTogglePropertyState from the Command5 drop-down list.
38. Click the Ellipsis button on the right of the Command5 field to display its associated Current Command tab and click this tab.
39. Here, since you want to enable or disable the DisplayByType property's graphic effects, click Enabled from the Status drop-down list so the property graphic effects are turned on or off according to the current property status. Click StationNodes in the Element Type1 field and click the related property, DisplayByType, for the Property1 field.



40. Save the specified parameters.

If you want to test this new context menu display, switch to Run mode, open a diagram, and right-click any StationNodes schematic node contained in this diagram. The context menu should display as in the following screen shot.

Note because this behavior is defined for the StationNodes element type, it is also available from the inherited StationNodes_F element type; that is, this context menu also displays when right-clicking a StationNodes_F node in any diagram implemented by the Feeder diagram type.



Note that for the last three menu items, the item is checked when the related property's effects are enabled and unchecked when they are disabled. Clicking one of these three context menus when they are checked deactivates the related property, while clicking one of these three context menus when they are unchecked activates the related property.

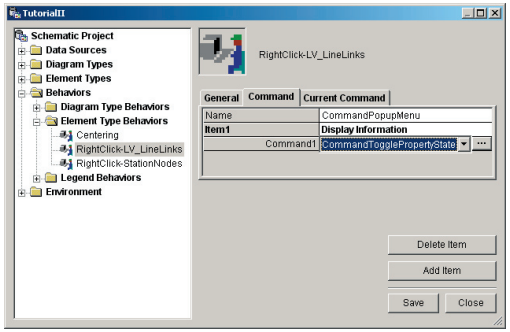
Defining the sample behavior related to the LV_LineLinks type

As you did for the StationNodes element type, you must now create a new behavior to display a context menu from any right-clicked LV_Lines link.

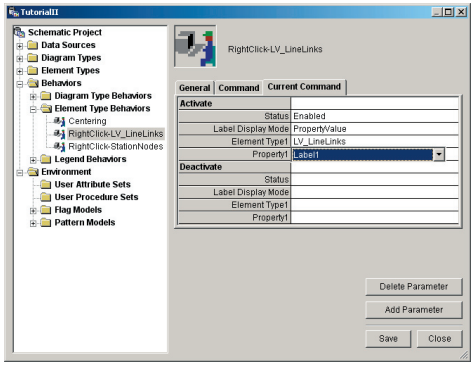
1. Right-click the Element Type Behaviors entry in the Designer tree and click Create.
2. From the Create Behavior dialog box, choose RightClick from the Event Name drop-down list.
   Because the Create Behavior dialog box has been opened from the Element Type Behaviors entry, Schematic Element should already be selected for Meta Type.
3. Then, click LV_LineLinks from the Object Type drop-down list and validate.
4. Click OK.
5. Now, from the newly created behavior's General tab, click CommandPopupMenu from the Command drop-down list.
6. Click the Command tab where you have to specify the menu items.

The first menu item must activate or deactivate the Label1 property label display:

7. Type the Item1 value (Display Information, in this example).
8. Click CommandToggleProperty from the Command1 drop-down list.



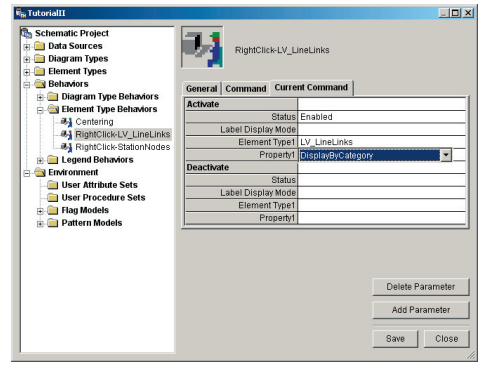
9. Click the Ellipsis button on the right of the Command1 field to display its associated Current Command tab and click this tab.
10. Here, because you want to enable or disable the Label1 property label display, click PropertyValue from the Label Display Mode drop-down list so the property label display is turned on or off.
11. Then, click LV_LineLinks from the Element Type1 drop-down list.
12. Click Label1 for the Property1 drop-down list.



Save the specified parameters.

You must now define the second context menu item that will be used to activate or deactivate the DisplayByCategory property's graphic effects.

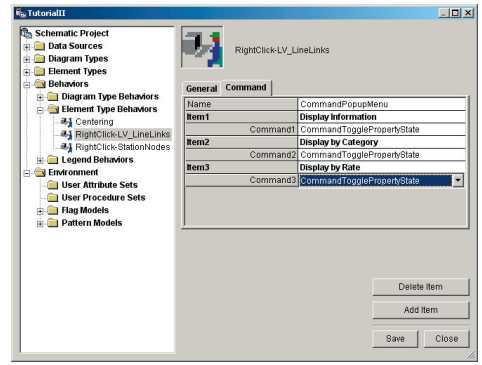
13. Click the Command tab and click Add Item to set the second context menu item.
14. Type the second item that will appear in the context menu (Display by Category in this example).
15. Click CommandTogglePropertyState from the Command2 drop-down list.
16. Click the Ellipsis button on the right of the Command2 field to display its associated Current Command tab and click this tab.
17. Since you want to enable or disable the DisplayByCategory property's graphic effects, click Enabled from the Status drop-down list; click LV_LineLinks in the Element Type1 field; and click the related property, DisplayByCategory, for the Property1 field.



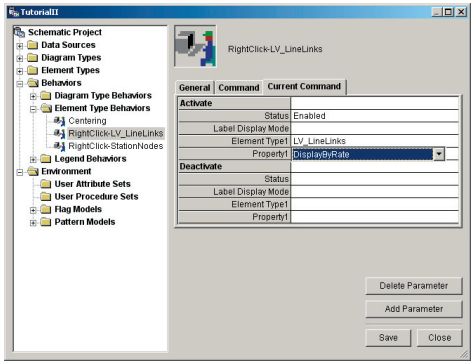
Save the specified parameters.

It only allows you to create the last item menu that must activate or deactivate the DisplayByRate property's graphic effects.

18. Repeat steps 13 through 15 so the Command tab content displays as follows:



19. Then, repeat steps 16 and 17 for the DisplayByRate property. The Current Command tab should display as follows:



20. Click Save.

If you switch to Run mode and test this new context menu display, it should appear as follows: