



# **IBM ILOG JViews Diagrammer V8.6**

## **Using the Designer**

# Copyright

## Copyright notice

© Copyright International Business Machines Corporation 1987, 2009.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

## Trademarks

IBM, the IBM logo, ibm.com, WebSphere, ILOG, the ILOG design, and CPLEX are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

## Notices

For further copyright information see `<installdir>/license/notices.txt`.

---

## *Table of contents*

<b>Getting started.....</b>	<b>9</b>
<b>Running the Designer.....</b>	<b>11</b>
<b>Creating a new diagram.....</b>	<b>13</b>
Launching the New Diagram Wizard.....	14
Choosing the format of data to import.....	15
Importing your JViews Diagrammer XML data.....	16
Editing the data model.....	17
Selecting the type of diagram.....	19
Choosing a graph layout theme.....	21
<b>Saving the diagram.....</b>	<b>23</b>
<b>Adding to the style sheet.....</b>	<b>24</b>
<b>Customizing genealogy nodes by type.....</b>	<b>27</b>
Creating a person rule.....	29
Renaming a rule.....	31
Assigning a specific symbol to the person nodes.....	32
Setting properties in the Styling Customizer for person nodes.....	34
Setting properties for females.....	36
Adding a death fact to person nodes.....	39
Customizing the death date.....	40
Creating a couple rule.....	41
Assigning a specific symbol to the couple nodes.....	42
Setting properties in the Styling Customizer for couple nodes.....	43

Rule-select and yellow icons.....	45
Status icons and local properties.....	46
<b>Customizing genealogy links by type.....</b>	<b>47</b>
Creating a couplelink rule.....	48
Setting couplelink properties.....	49
<b>Testing application behavior.....</b>	<b>51</b>
<b>Adding and customizing a legend.....</b>	<b>52</b>
<b>Integrating your development into an application.....</b>	<b>54</b>
<b>Reusing a project in the Designer.....</b>	<b>56</b>
<b>Getting to know the Designer.....</b>	<b>57</b>
<b>Style rules.....</b>	<b>59</b>
Style Rules pane.....	61
Options.....	62
Node and link rules.....	64
Level of detail.....	65
Styling customizers.....	66
Styling properties.....	67
Priority.....	68
Style sheet debugging.....	69
Resetting of default property values.....	70
<b>Graph and link layouts.....</b>	<b>71</b>
<b>Label layout.....</b>	<b>72</b>
<b>Symbols.....</b>	<b>73</b>
About symbols.....	74
Styling Customizer.....	75
Assigning a symbol.....	76
Editing a symbol.....	78
Refreshing symbols.....	79
Palette management.....	80
<b>General links.....</b>	<b>81</b>
<b>Link connections.....</b>	<b>82</b>
<b>Interactors.....</b>	<b>83</b>
About interactors.....	84
Predefined interactors.....	85
Customizing interactors.....	87
Customization options.....	88
Linking predefined interactors to parameters.....	89
Controlling interactions using CSS functions.....	91
Customizing the diagram dynamically.....	92

<b>Working in different modes.....</b>	<b>97</b>
The three modes.....	98
Working in Style Editing mode.....	99
Working in Preview mode.....	101
Working in Diagram Editing mode.....	102
<b>Using wizards.....</b>	<b>107</b>
New Diagram wizard.....	108
Create Style Rule wizard.....	109
Change Style Rule wizard.....	112
Graph Layout wizard.....	113
Data wizard.....	114
<b>The Rule menu.....</b>	<b>115</b>
<b>The Undo and Redo facilities.....</b>	<b>118</b>
<b>The Check Rules facility.....</b>	<b>119</b>
<b>Messages for troubleshooting.....</b>	<b>120</b>
<b>Printing the diagram.....</b>	<b>121</b>
<b>Comparing the Designer and the API.....</b>	<b>124</b>
<b>Using more Designer features.....</b>	<b>125</b>
<b>Changing Layout Parameters.....</b>	<b>127</b>
Customizing graph layout algorithms.....	128
Using the genealogy diagram.....	129
Changing a basic parameter.....	130
Changing a link style parameter.....	131
Setting a link style with mixed links.....	132
<b>Using the Graph Layout Wizard.....</b>	<b>134</b>
<b>Loading a different style sheet.....</b>	<b>136</b>
<b>Resetting a default value after a state change.....</b>	<b>137</b>
<b>Importing data from a flat file.....</b>	<b>139</b>
Loading the file.....	140
Importing node data.....	141
Importing link data.....	142
Mapping node data.....	143
Mapping link data.....	145
Editing the data model.....	146
Selecting the diagram type.....	148
<b>Inserting an icon.....</b>	<b>151</b>
Image file formats.....	152
Setting a special symbol.....	153
Creating rules for population ranges.....	154

Rule priority.....	156
<b>Creating a user-defined data model.....</b>	<b>157</b>
Creating the data model.....	158
Editing the data model.....	159
Changing the graphic representation of the nodes.....	163
<b>Adding decorations for alarm severities.....</b>	<b>164</b>
<b>Adding a node to a diagram.....</b>	<b>166</b>
<b>Using subgraphs.....</b>	<b>169</b>
Creating subgraphs by grouping objects.....	170
Styling subgraphs.....	172
<b>Displaying different levels of detail.....</b>	<b>174</b>
<b>Using background maps.....</b>	<b>176</b>
<b>Importing data from a database.....</b>	<b>177</b>
Loading the database.....	178
Connecting to the database.....	179
Mapping the database.....	182
Editing the data model.....	186
Selecting the diagram type.....	188
Choosing a Graph Layout Theme.....	189
<b>Using data model expressions.....</b>	<b>191</b>
About references and expressions.....	192
Using expressions in text properties.....	193
Using expressions in numeric or fixed-value properties.....	194
Using Expression Assist.....	195
Using variables.....	196
Using custom functions.....	197
Displaying help on expressions.....	199
<b>Writing an application.....</b>	<b>201</b>
<b>Deployment considerations.....</b>	<b>202</b>
<b>JViews Diagrammer classes available as beans.....</b>	<b>203</b>
<b>Creating a JViews Diagrammer application.....</b>	<b>205</b>
Creating an application instance.....	206
Customizing the predefined application.....	207
Creating a basic application from JViews Diagrammer beans.....	208
Adding further components.....	210
An editing application created from JViews Diagrammer beans.....	211
<b>Printing the diagram from an IlvDiagrammerApplication instance.....</b>	<b>212</b>
<b>Data formats.....</b>	<b>213</b>

<b>About supported formats.....</b>	<b>214</b>
<b>Application data in XML files for JViews Diagrammer.....</b>	<b>215</b>
<b>Application data in flat files for JViews Diagrammer.....</b>	<b>220</b>
<b>Application data in databases for JViews Diagrammer.....</b>	<b>222</b>
<b>Project files.....</b>	<b>224</b>
<b>Style sheet files.....</b>	<b>225</b>
<b>Next steps after the Designer.....</b>	<b>226</b>
<b>Index.....</b>	<b>229</b>





# Getting started

Explains how to use the Designer for JViews Diagrammer to start developing a JViews Diagrammer diagram component for your application, based on an example: a genealogical tree.

## In this section

### Running the Designer

Explains how to launch the Designer under Windows® and under UNIX® .

### Creating a new diagram

Explains how to create a diagram through the New Diagram Wizard.

### Saving the diagram

Describes the files that are created when you save the newly created diagram.

### Adding to the style sheet

Explains how to customize the appearance of a diagram through style rules.

### Customizing genealogy nodes by type

Explains how to customize the appearance of a `person` node and make it look different from a `couple` node.

### Customizing genealogy links by type

Explains how to make a `couple` link look different from a `child` link.

### Testing application behavior

Explains how to switch to Preview mode to test your changes in an application.

**Adding and customizing a legend**

Explains how to add a legend to the diagram.

**Integrating your development into an application**

Explains the different ways of integrating the diagram into an application.

**Reusing a project in the Designer**

Explains the conditions under which you can reuse a project in the Designer.

---

## Running the Designer

You can run the Designer from Windows® or UNIX® operating systems. The startup conditions are the same in both cases.

### To run the Designer under Windows

- ◆ In the Start menu, select **All Programs>IBM ILOG>IBM ILOG JViews Diagrammer 8.6>Designer**.

IBM® ILOG® is the default program group and may be different if you have installed JViews Diagrammer into a different group.

-- or --

1. Go to the directory `<installdir>/jviews-diagrammer86/bin/designer`.
2. Double-click `run.bat`.

### To run the Designer under UNIX

1. Go to the directory `<installdir>/jviews-diagrammer86/bin/designer`.

Make sure your PATH environment variable is set correctly to find this directory.

2. Enter `run.sh`.

When you start the Designer, it opens with the New Diagram Wizard displayed as if you had chosen **File>New from Wizard**. You can close the wizard if you do not want to create a new diagram as proposed. You can also enable or disable the opening of the wizard on startup from its first page.

Outside the wizard, the way the Designer operates depends on the mode it is in. You can see the current mode in the title bar, and you can change the mode using the buttons in the vertical toolbar. For a discussion of modes, see *Working in different modes*.



# *Creating a new diagram*

Explains how to create a diagram through the New Diagram Wizard.

## **In this section**

### **Launching the New Diagram Wizard**

Explains how to launch the wizard that helps you create a new diagram.

### **Choosing the format of data to import**

Lists the different choices of data formats and how to select the one to use in this tutorial.

### **Importing your JViews Diagrammer XML data**

Explains how to import XML data from which to create a diagram.

### **Editing the data model**

Explains how to edit the XML file data model.

### **Selecting the type of diagram**

Presents the choices of diagram and how to select the appropriate one for your diagram.

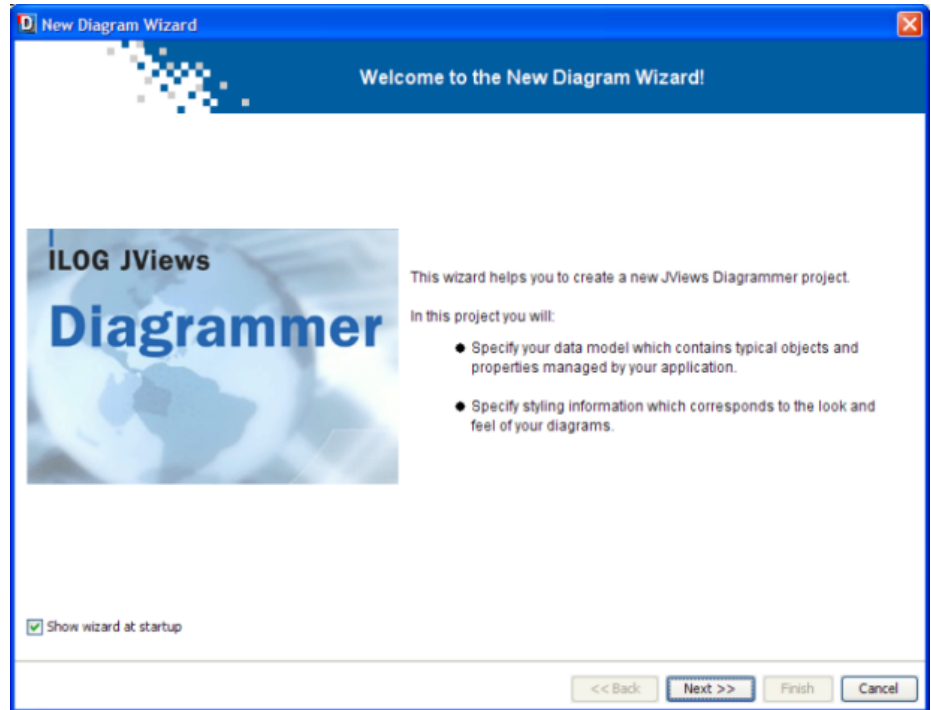
### **Choosing a graph layout theme**

Proposes a list of graph layout choices and explains how to select the one for this tutorial.

## Launching the New Diagram Wizard

To start a new diagram if the wizard is not open:

1. Choose **File>New from Wizard**.



*New Diagram Wizard*

2. Click **Next** to choose the type of data source.

## Choosing the format of data to import

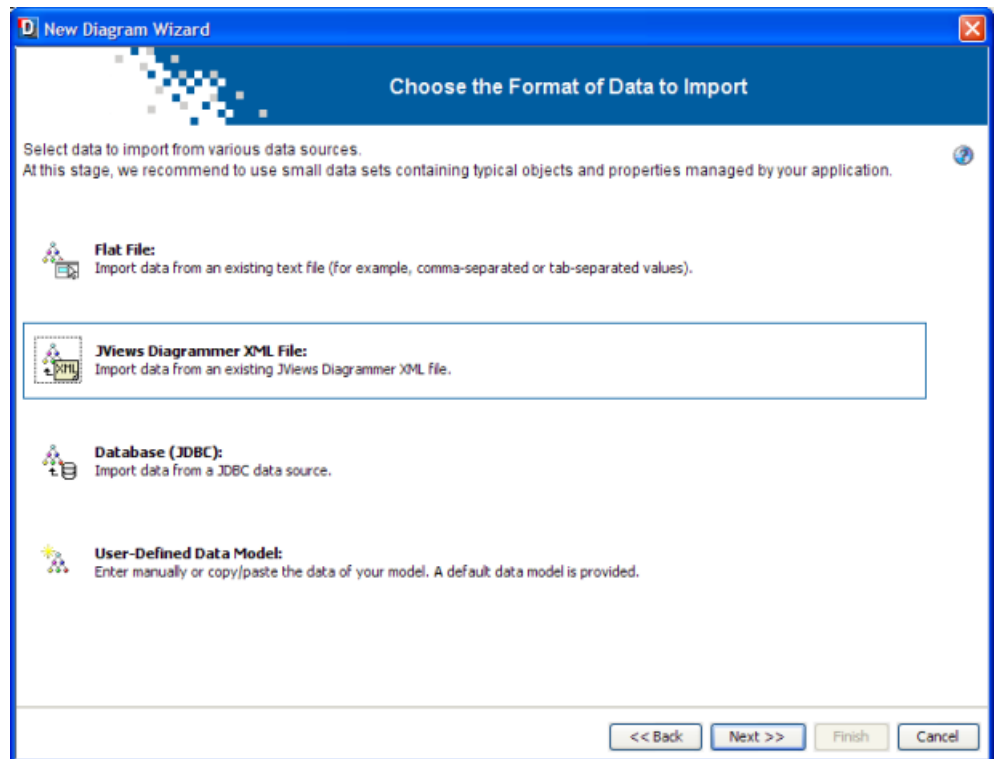
The New Diagram Wizard gives you a choice of the following formats of data to import:

- ◆ Flat file, for example, Microsoft® Excel® file in CSV (comma-separated values) format
- ◆ XML file in diagram format
- ◆ Database supported by JDBC™ (Java Database Connectivity), for example, Microsoft® Access®
- ◆ User-defined model, to start from a default data model with a few data items

The example built up in this section makes use of data in an XML file.

For an example of a database, see *Importing data from a database*. For an example of a flat file, see *Importing data from a flat file*. For an example of the user-defined option, see *Creating a user-defined data model*.

1. Select **XML file**.
2. Click **Next**.



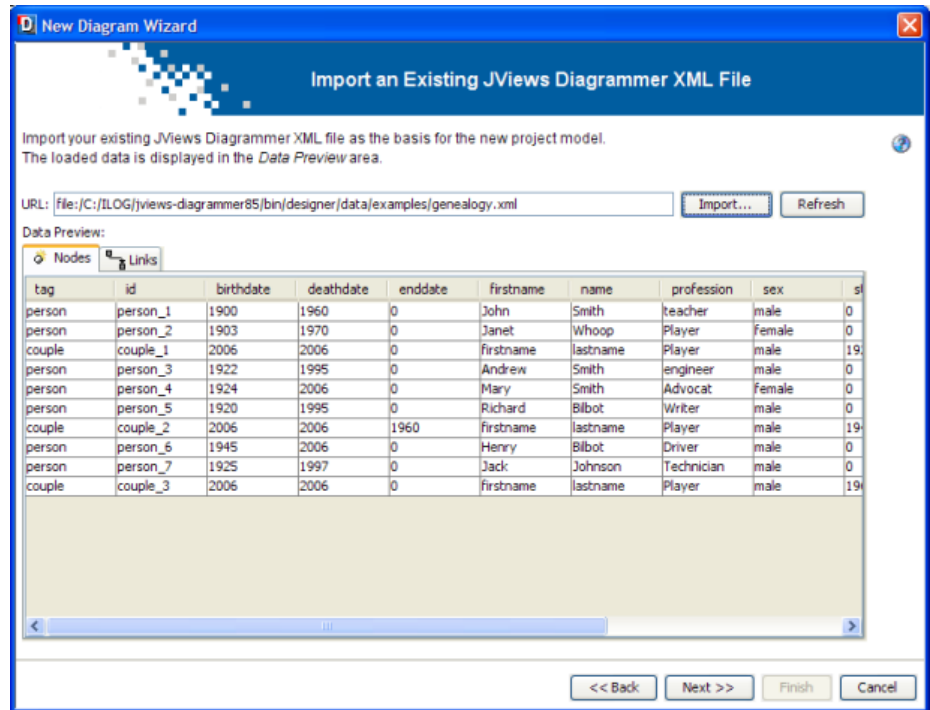
*Choosing the XML File option*

## Importing your JViews Diagrammer XML data

An XML document is supplied as file `genealogy.xml` in the directory `<installdir>/jviews-diagrammer86/bin/designer/data/examples` of JViews Diagrammer. This document contains a data model for a genealogy example.

### To load data from the file `genealogy.xml`:

1. Click the Import button.
2. Browse to the directory `<installdir>/jviews-diagrammer86/bin/designer/data/examples` of JViews Diagrammer.
3. Select the XML file `genealogy.xml`.
4. Click Open.



### Choosing to load data from an XML file

You see your data in table format: a table for the nodes and a table for the links. Note that the XML data includes x and y values in the genealogy example.

5. Click Next to move to the Edit Your Data Model page.



---

## Editing the data model

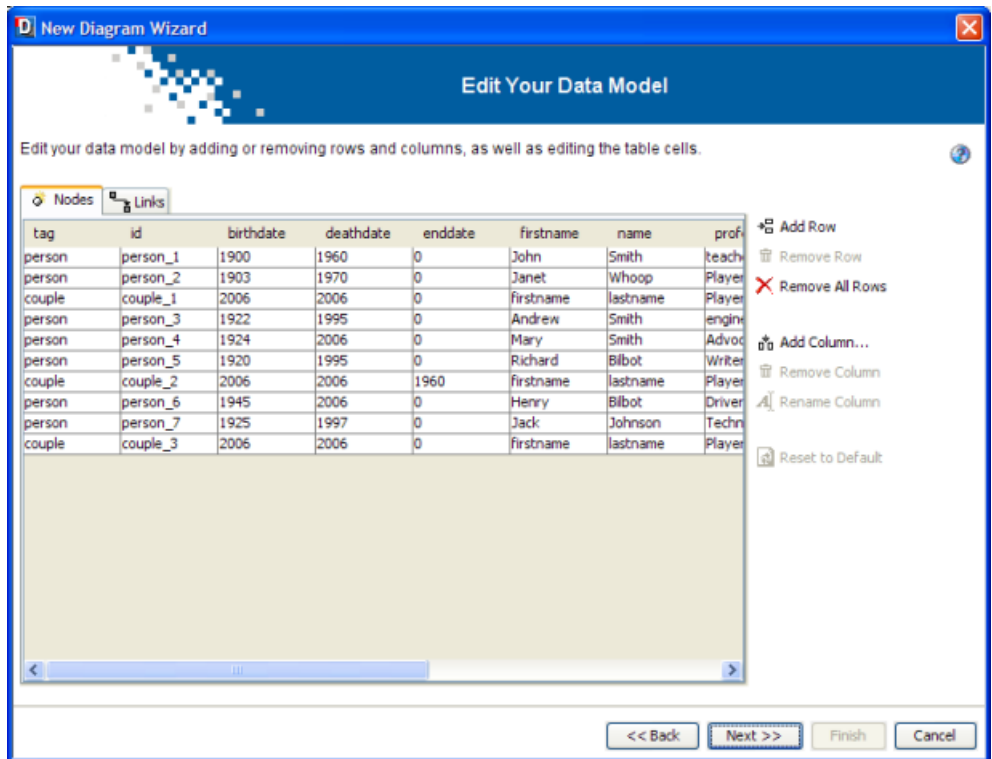
You can edit the basic data model to:

- ◆ Add and remove rows (a row represents an object in the data model)
- ◆ Add, remove, rename columns (a column represents a property of the objects in the data model)
- ◆ Change cell values (a cell represents a property of a specific object)
- ◆ Remove all objects
- ◆ Restore the data model to its initial state
- ◆ Paste data from the clipboard

If you click **Remove All Rows**, you clear all the data values. To retrieve the values you saw initially, click the **Reset to Default** button.

Properties `id` and `tag` are mandatory for the SDM model and therefore cannot be removed or renamed by the user.

**Paste from Clipboard** replaces the contents of the current table by the information present in the clipboard. It asks if the user wants the column names to be named based on the first row or not. If so, the first line of data is used to give the column names. If not, the column names are named as A, B, C, and so on, and can later on be renamed by the user manually.



### *Editing the XML file data model*

You will keep this data model as it is.

#### **To validate your data model:**

- ◆ Click **Next**.

---

## Selecting the type of diagram

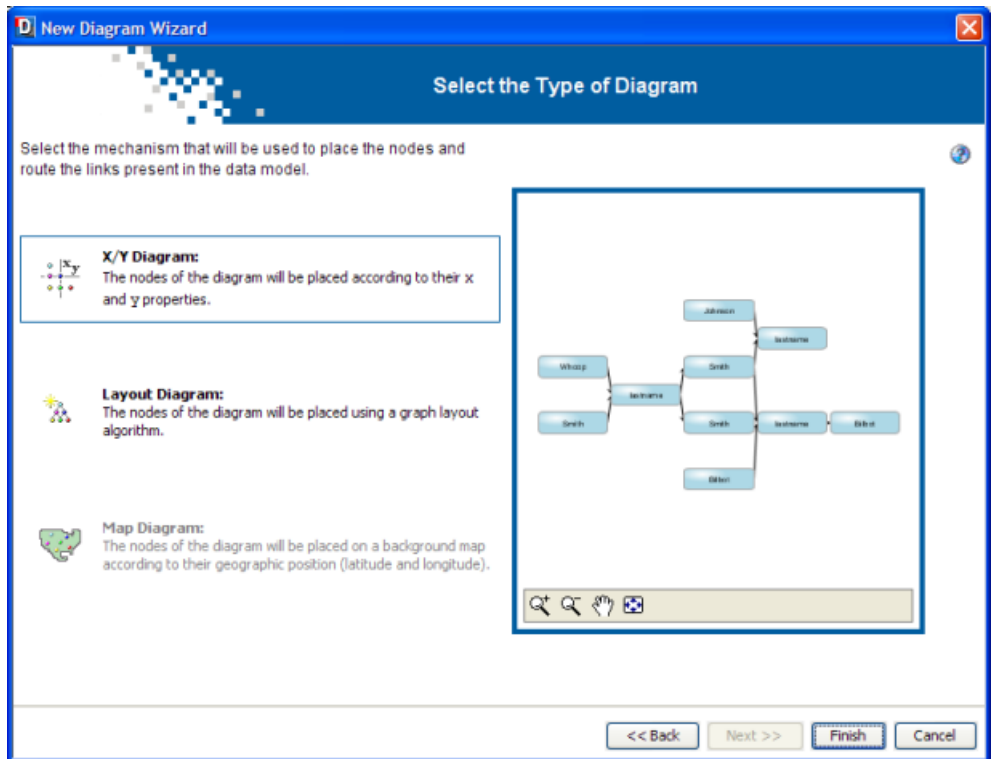
When you load data from an XML file, a default style is automatically applied to the nodes and links. You can see a preview in the small pane on the right of the wizard.

You can select from three types of diagram:

- ◆ **X/Y Diagram:** Nodes are positioned according to x- and y-coordinates. The x and y attributes must be present in the data.
- ◆ **Layout Diagram:** Nodes are positioned according to a graph layout algorithm. The default graph layout algorithm is Hierarchical.
- ◆ **Map Diagram:** Nodes are positioned according to their latitude and longitude values on a map. The latitude and longitude attributes must be present in the data.

**Note:** You can only build a Map Diagram if you have purchased and installed the IBM® ILOG® JViews Maps product. The Designer warns you if JViews Maps is not available when it is needed.

The XML file contains x and y values and so the X/Y Diagram is selected by default, see the following figure.



### *Specifying a diagram type*

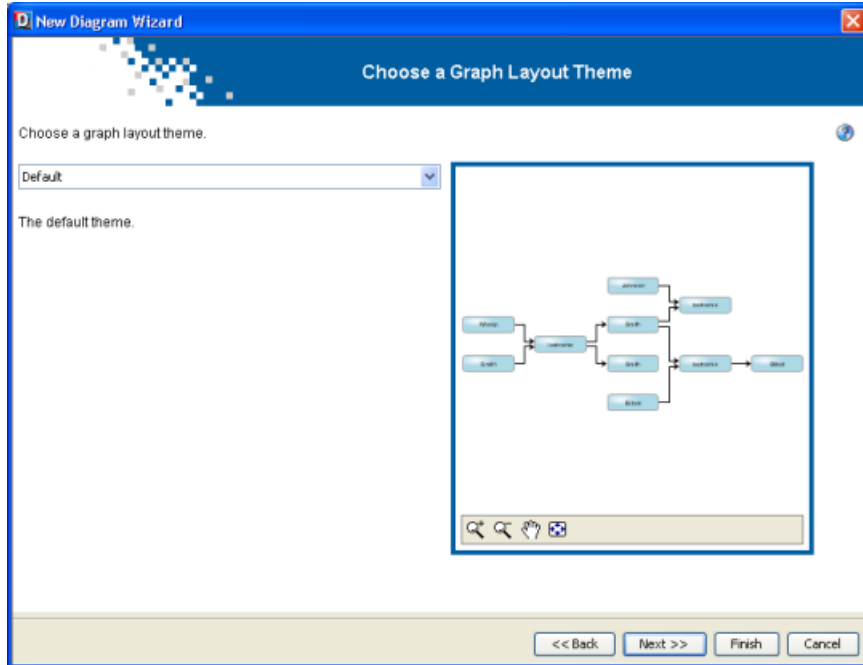
If positioned according to their x and y coordinates, the nodes overlap. A graph layout gives better results.

#### **To select a different type of diagram:**

1. Click the icon for Layout Diagram.
2. Click Next.

## Choosing a graph layout theme

A graph layout theme applies styling to a graph layout to configure the way nodes and links are interconnected and arranged. A layout theme consists of a partial style sheet. It is a convenient way of applying a precustomized layout that you want to reuse in different projects.



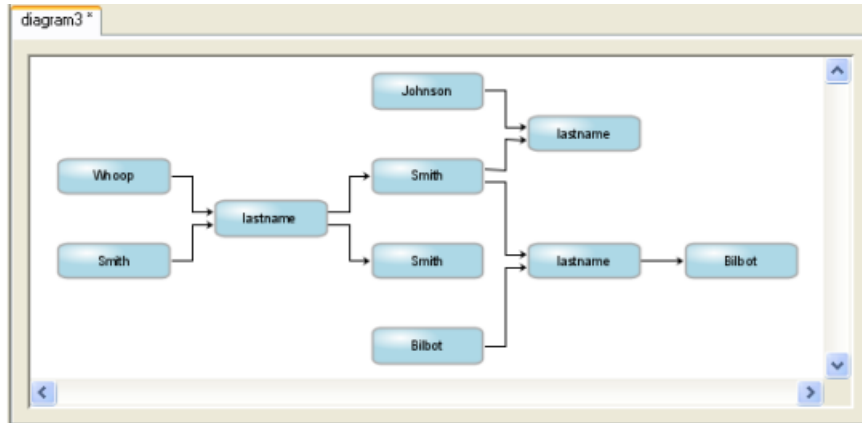
### *Choosing a layout theme*

For this example, you can leave the default layout theme.

- ◆ Click Finish.

If you need to make changes, you can click Back instead to return to specific wizard pages.

When you exit the wizard, your diagram is displayed in the Diagram pane, see the following figure.



*The initial view of the genealogy diagram*

You can now change the styling as required in Style Editing Mode.

---

## Saving the diagram

If you do not want to lose your work, you must not forget to save your newly created diagram.

### To save the diagram initially:

1. Choose **File>Save As**.
2. Enter a file name in the file chooser.

After saving the first time and if you have made further changes, you can just choose **File>Save** or click Ctrl+S to save again.

When you save your work, the Designer saves three files: a project file, a style sheet, and a data file.

The following example shows a project file for an XML data source.

### A JViews Diagrammer project file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE diagrammer>
<diagrammer style="genealogy.css">
  <datasource class="ilog.views.diagrammer.datasource.IlvXMLDataSource"
url="genealogy.xml" />
</diagrammer>
```

The project file specifies the name of the style sheet file, the type and parameters of the data source, and the name of the data file. You can see that this project file is for the genealogy example.

If you try to open a project file while its style sheet and/or data file are missing, you will get an explicit error message prompting you to specify the location of the missing files. Keep in mind that the three files must live together.

---

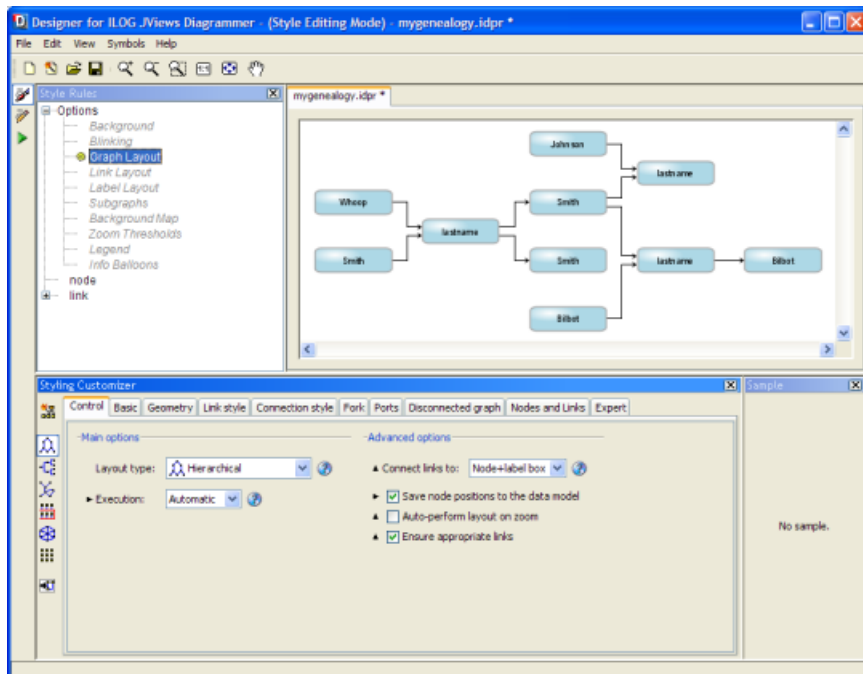
## Adding to the style sheet

JViews Diagrammer uses cascading style sheets as a powerful tool to create and customize graphic objects that represent the objects in your data model.

When you start a diagram through the wizard and load data from an XML file, default styling is applied.

The default theme gives a plain appearance and does not show all the data, see figure *Initial diagram and graph layout customizer*. You can change this by defining your own style rules.

The graph layout used is Hierarchical, from left to right. This is governed by the Graph Layout option in the Options section of the Style Rules. You can see the Styling Customizer if you select Graph Layout, see the following figure.



*Initial diagram and graph layout customizer*

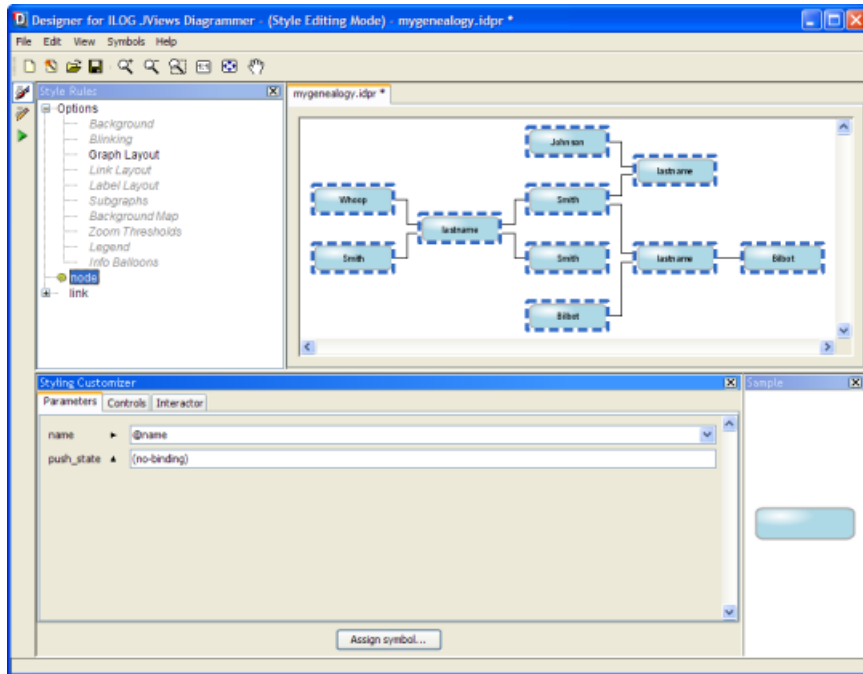
You can see all the default style rules in a tree structure. You can add a new rule, modify an existing rule, delete an existing rule, and change the priority of a rule (its level in the list) using the Rule menu which is available when you right-click a rule in the tree.

---

## Selecting the rule on all nodes

There is one rule defined for all nodes in the diagram. This rule sets the default shape, size, coloring, and label of all nodes. When you select the `node` entry in the Style Rules, all nodes to which this general rule applies are selected in the diagram and you can see their properties in the Styling Customizer, as shown in the following figure.





### Selecting the node rule

The nodes in the diagram are represented by symbols.

Note the yellow icon next to the `node` rule; it indicates that the rule applies to the selected objects. Sometimes there is more than one yellow icon displayed, see *Rule-select and yellow icons*.

The most general rule (`node`) applies to all nodes. Other node rules are more specific than this one.

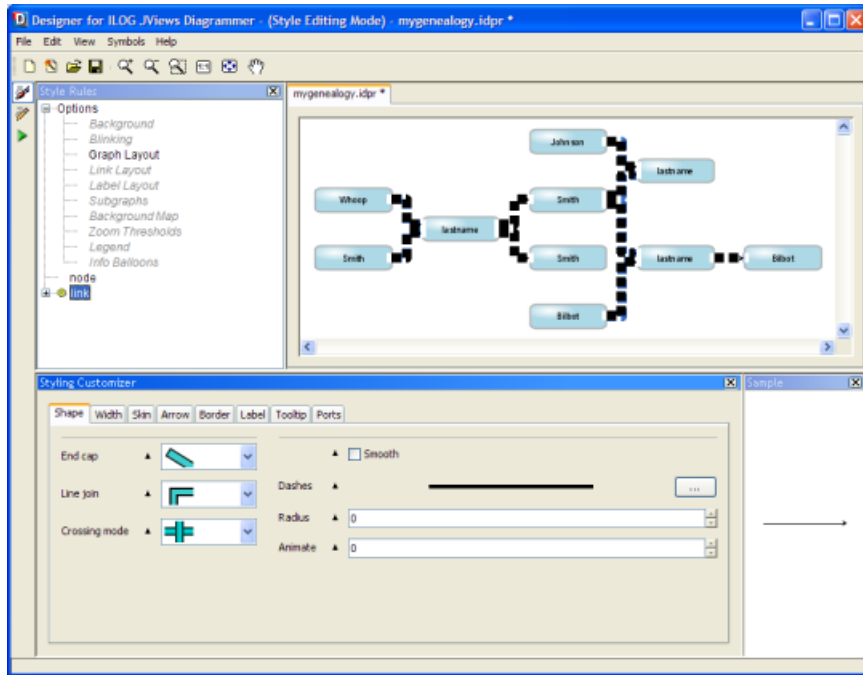
In the Style Rules, you can set a rule on nodes in general by selecting `node`. There can only be one rule at the top level, that is, one `node` rule and one `link` rule. If you try to create a second `node` rule or `link` rule, you are prompted to delete the existing one. You can think of these general rules as setting default property values for all nodes and all links.

The data model specified in the XML file contains nodes and links of the following user-defined types: `person`, `couple`, `couplelink`, and `childlink`. These types are the top-level elements in the XML file. You can set a rule on any nonlink top-level element in the XML file by selecting `node` and then specifying a valid user-defined type, for example, `node.person`. See *Customizing genealogy nodes by type* to learn how to apply specific rules to nodes.

---

## Selecting the rule on all links

There is one rule defined for all links in the diagram. This rule sets the default line width, coloring, and arrowhead (if any) of all links. When you select the `link` entry in the Style Rules, all links to which this general rule applies are selected in the diagram, and the properties set are displayed in the Styling Customizer, see the following figure.



### Selecting the link rule

The links in the diagram are represented by a general link object, which is customized in terms of its Shape and Skin (content), and optional Arrow, Border, and Label.

The `link` rule has a yellow icon next to it because it applies to the selected objects (all the links).

In the Style Rules, you can set a rule on links in general by selecting `link`.

The data model specified in the XML file contains nodes and links of the following user-defined types: `person`, `couple`, `couplelink`, and `childlink`. These types are the top-level elements in the XML file. You can set a rule on any link element in the XML file by selecting `link` and then specifying a valid type of link, for example, `link.childlink`. See *Customizing genealogy links by type* to learn how to apply specific rules to links.

# *Customizing genealogy nodes by type*

Explains how to customize the appearance of a `person` node and make it look different from a `couple` node.

## **In this section**

### **Creating a person rule**

Explains how to define nodes as persons through style rules.

### **Renaming a rule**

Explains how to customize the name of a rule.

### **Assigning a specific symbol to the person nodes**

Explains how to change the graphic representation of the person nodes.

### **Setting properties in the Styling Customizer for person nodes**

Explains how to customize each node to display a specific name and a birthdate.

### **Setting properties for females**

Explains how to visually distinguish male from females person nodes.

### **Adding a death fact to person nodes**

Explains how to identify dead persons in the diagram.

### **Customizing the death date**

Explains how to customize the death date.

### **Creating a couple rule**

Explains how to define nodes as couples through style rules.

**Assigning a specific symbol to the couple nodes**

Explains how to change the graphic representation of the couple nodes.

**Setting properties in the Styling Customizer for couple nodes**

Explains how to customize each node to display a specific status and a date.

**Rule-select and yellow icons**

Explains how to interpret the visual feedback you get when selecting a rule.

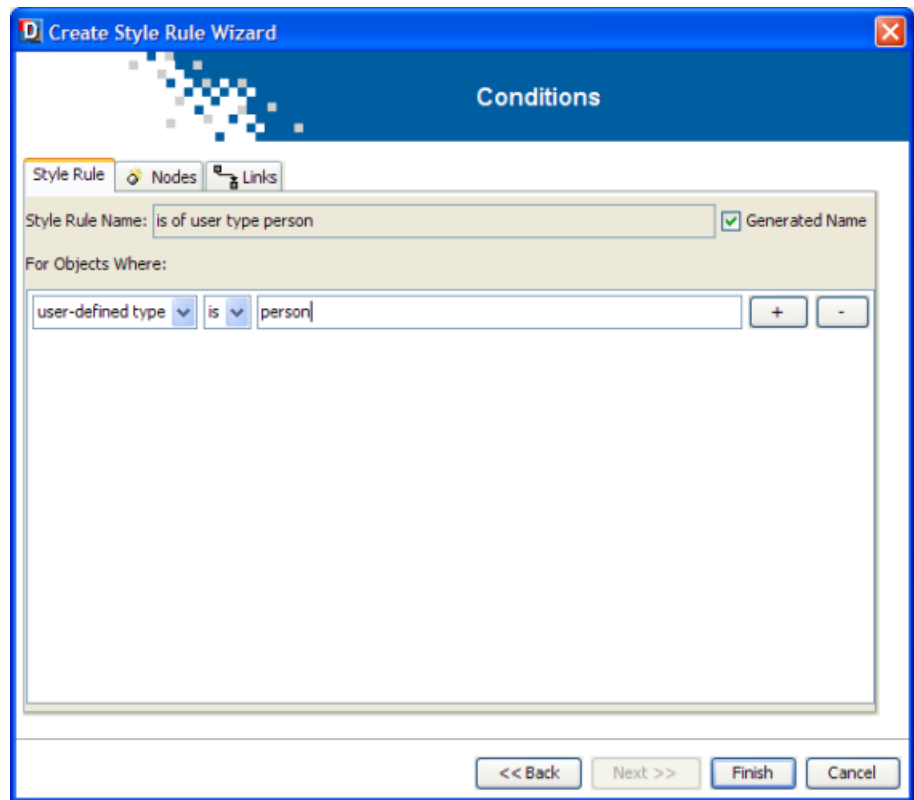
**Status icons and local properties**

Explains how to distinguish local and inherited properties.

## Creating a person rule

### To create a person rule:

1. Select the `node` rule in the Style Rules.
2. Choose **Edit>Create Style Rule**.  
The Create Style Rule Wizard opens.
3. Click Next to reach the Conditions page.
4. Select `user-defined type` in the first drop-down list.
5. Type `person` in the field next to the second drop-down list. See the following figure.



*Creating a new style rule on a user-defined type*

6. Click Finish to exit the wizard.

The new rule appears in the Style Rules for nodes and is selected.

When you add a more specific rule, like the `is of user type person` rule, it overrides some or all of the property values set by the general rule for the objects affected.

**Note:** If you do not specify a rule name, the default name will be concatenated from the condition strings.

---

## Renaming a rule

You can rename the rule at any time in two different ways.

**To rename a rule:**

1. In the Style Rules pane, right-click the rule.
2. Choose **Rename** in the shortcut menu.
3. Type `node.person` in the dialog box.
4. Click OK.

The new rule name displays in the Style Rules pane.

-- or --

1. With the rule selected, choose **Change Style Rule**.
2. Uncheck the Generated Name box.
3. Type the new name in the Style Rule Name field.
4. Rename the rule to `node.person`.

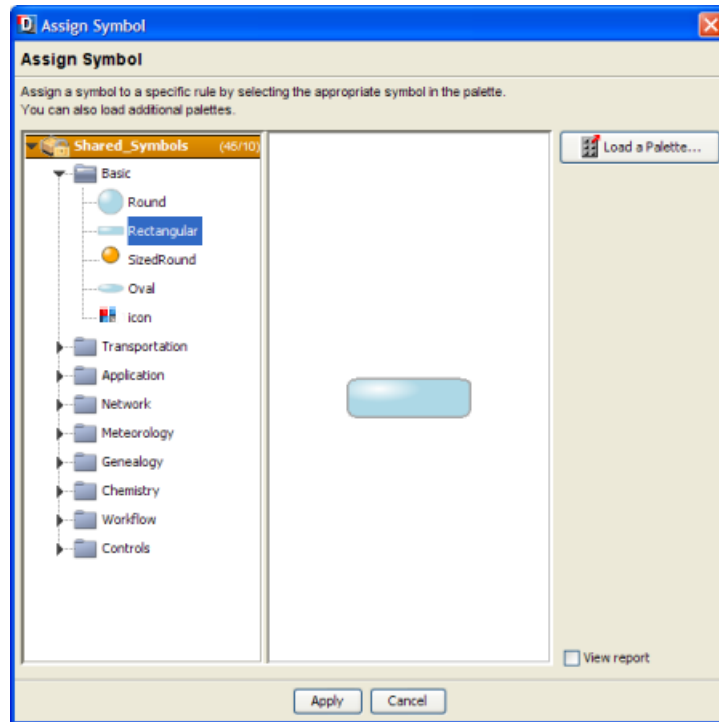
## Assigning a specific symbol to the person nodes

The symbol used in the diagram is quite basic. You will assign a different symbol to the person nodes to graphically represent more data, such as the birthdate and the deathdate.

### To change the symbol assigned to person nodes:

1. Select the new `node.person` rule.
2. Click the Assign symbol button at the bottom of the Styling Customizer.

The Assign Symbol dialog box displays.

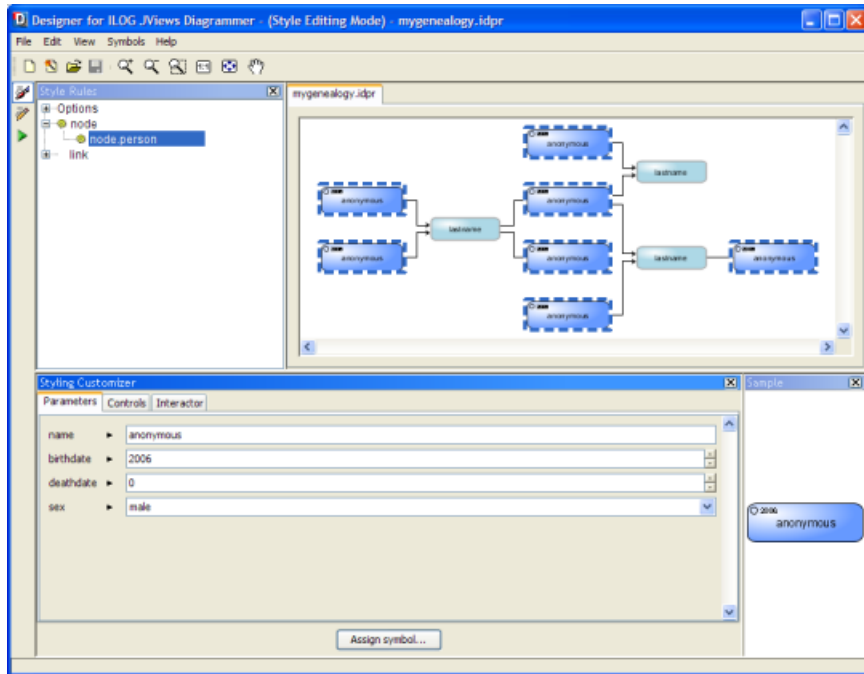


### *Assigning a symbol*

The `jviews-palette-shared-symbols-8.6.jar` palette is loaded by default. This palette is located in `<installdir>/jviews-diagrammer86/lib/palettes`.

3. Open the Genealogy category of symbols in the left pane.
4. Select the symbol Person.
5. Click Apply.





*Assigning the person symbol to nodes*

---

## Setting properties in the Styling Customizer for person nodes

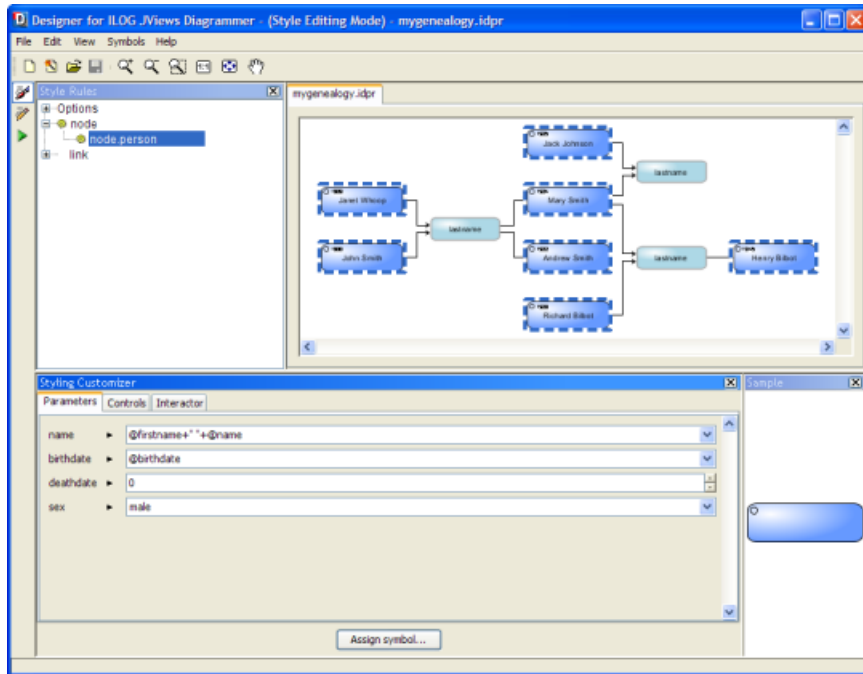
With the new `node.person` rule selected, you will set name and birthdate properties in the Styling Customizer.

### To set properties for the person nodes:

1. Select the `node.person` rule in the Style rules pane.
2. In the Parameters tab of the Styling Customizer, set name to `@firstname+" "+@name:`
  - ◆ Right-click in the field.
  - ◆ Choose Enter an Expression.
  - ◆ Delete the current value.
  - ◆ Select `@firstname` from the list of available expressions.
  - ◆ Type `" "+`.
  - ◆ Select `@name` from the list of expressions.
  - ◆ Press ENTER.

The commercial at character, @, is an operator specifying a model value: it prefixes the name of an attribute in the data model. Since some people have the same last name, change the value of the name parameter to `@firstname+" "+@name` and then press ENTER. The expression here concatenates the value of the `firstname` attribute, a space, and the value of the `name` attribute.
3. Set birthdate to `@birthdate` in the same way.
4. Press ENTER.

You can see the diagram change as you work, see the following figure.



*Customized person nodes*

**Note:** Remember that you can undo any unwanted changes in your Style Rules or Styling Customizer by choosing **Edit>Undo**.

---

## Setting properties for females

You can use the value of `sex` as either `male` or `female` to style subsets of person nodes differently. The color of a person node can be used to distinguish between `male` and `female`: blue for a man, red for a woman.

### To create a rule for females:

1. Select `node.person` in the Style Rules.
2. Choose **Edit>Create Style Rule**.

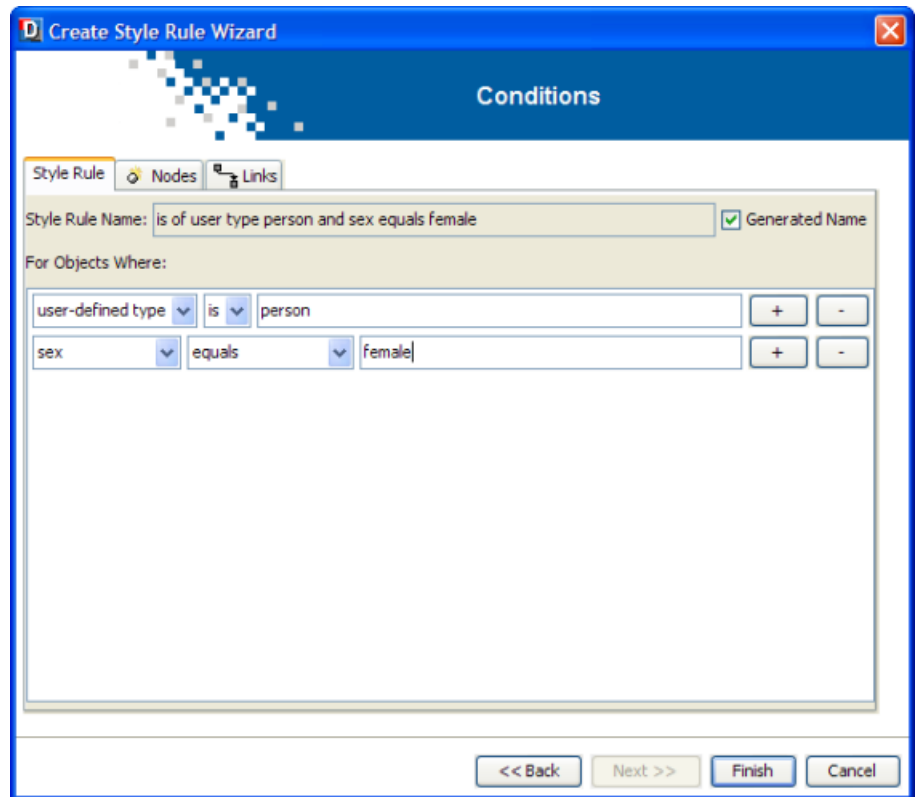
The Create Style Rule Wizard opens.

3. Click Next.

In the Conditions page, the condition `user-defined type is person` is already present as it has been defined for the `node.person` rule.

4. Click the plus button to add a new condition.
5. Select `sex` then `equals` and type `female`.

See the following figure.



*Creating a new data rule with an attribute condition*

6. Click Finish.

The new rule is in the Style Rules and is selected.

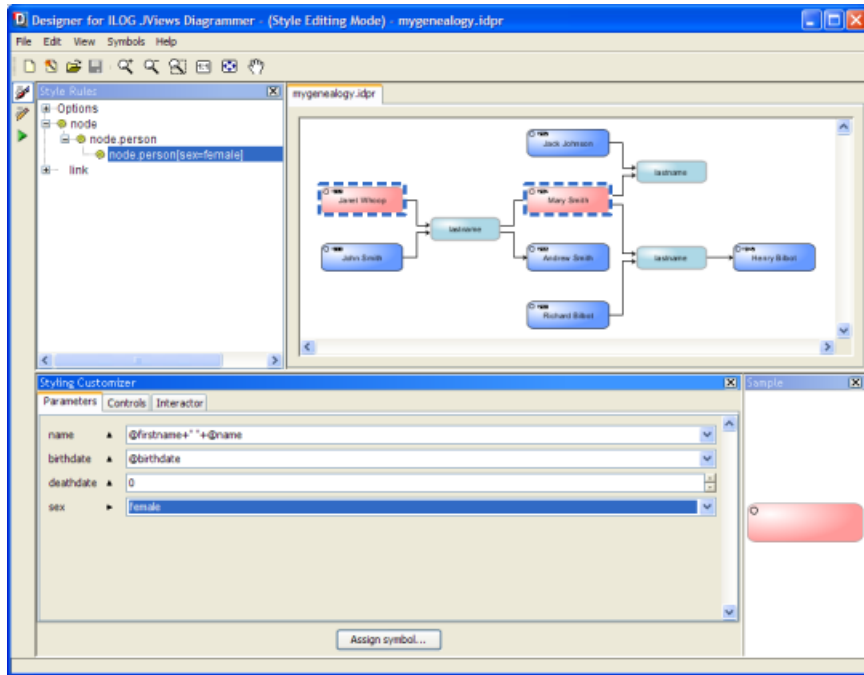
7. Right-click the rule name and rename it to something shorter: `node.person [sex=female]`.

**To set the female property:**

1. Select the `node.person [sex=female]` rule in the Style Rules pane.
2. In the Styling Customizer, select `female` for the `sex` parameter.

**Note:** Remember that you can undo any unwanted changes in your Style Rules or Styling Customizer by choosing **Edit>Undo**.

The result should be as shown in the following figure.



*Customized person nodes for men and women*

---

## Adding a death fact to person nodes

At this stage, the person nodes have a label showing the person's name as well as their birthdate. You will now add a death fact for the persons that are dead. This fact will not appear for living persons. Therefore you need to create a style rule to insert the death date only for dead persons.

### To add a death fact to person nodes:

1. Select the `node.person` rule.
2. Choose **Edit>Create Style Rule**.
3. Click Next.

In the Conditions page, the condition `user-defined type is person` is already present as it has been defined for the `node.person` rule.

4. Click the plus button to add a new condition.
5. Select the attribute `deathdate` and select `exists` and `not null`.
6. Click Finish.

The new rule is in the Style Rules and is selected.

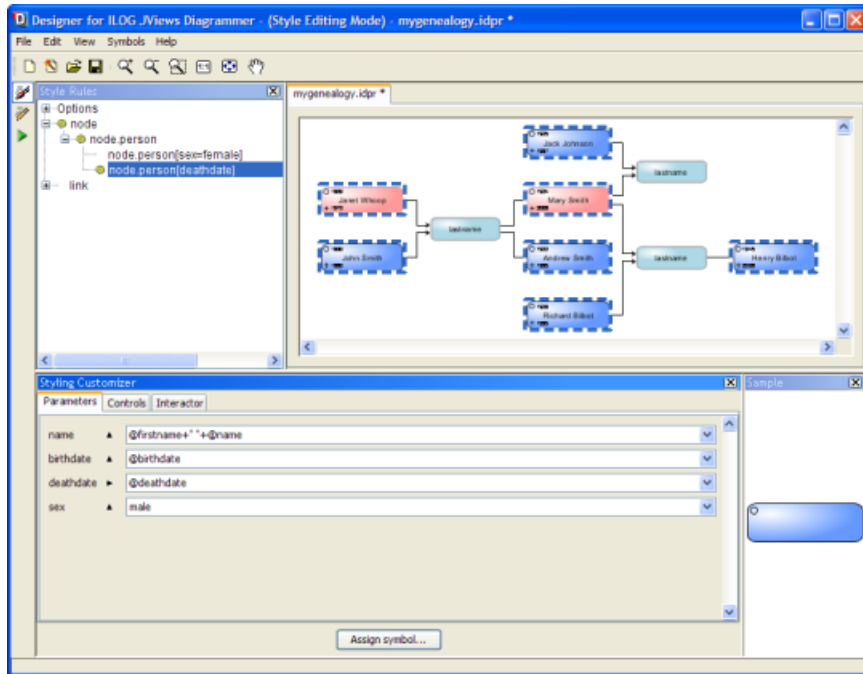
7. Right-click the rule name and rename it to something shorter: `node.person [deathdate]`.

## Customizing the death date

### To customize the death date:

1. Select the `node.person[deathdate]` rule in the Style Rules pane.
2. In the Styling Customizer, right-click in the `deathdate` field and choose Enter an expression.
3. Delete the current value and select `@deathdate` from the list of available expressions.
4. Press ENTER.

The result should be as shown in the following figure.



*Customizing the death date of person nodes*



---

## Creating a couple rule

### To create a couple rule:

1. Select the `node` rule in the Style Rules pane.
2. Choose **Edit>Create Style Rule.** \n  
The Create Style Rule Wizard opens.
3. Click Next to reach the Conditions page.  
The Conditions page of the Create Style Rule Wizard displays.
4. Uncheck Generated Name and type `node.couple` in the Style Rule Name field.
5. Select `user-defined type` in the first drop-down list, then type `couple` in the field next to the second drop-down list.
6. Click Finish to exit the wizard.

The new rule appears in the Style Rules for nodes and is selected.

**Note:** The check box Generated Name allows you to switch back at any time to the default name of a rule which is a concatenation of the condition strings.

## Assigning a specific symbol to the couple nodes

The symbol used in the diagram is quite basic. You will assign a different symbol to the couple nodes to graphically represent more data, such as the marital status and the date of the marriage.

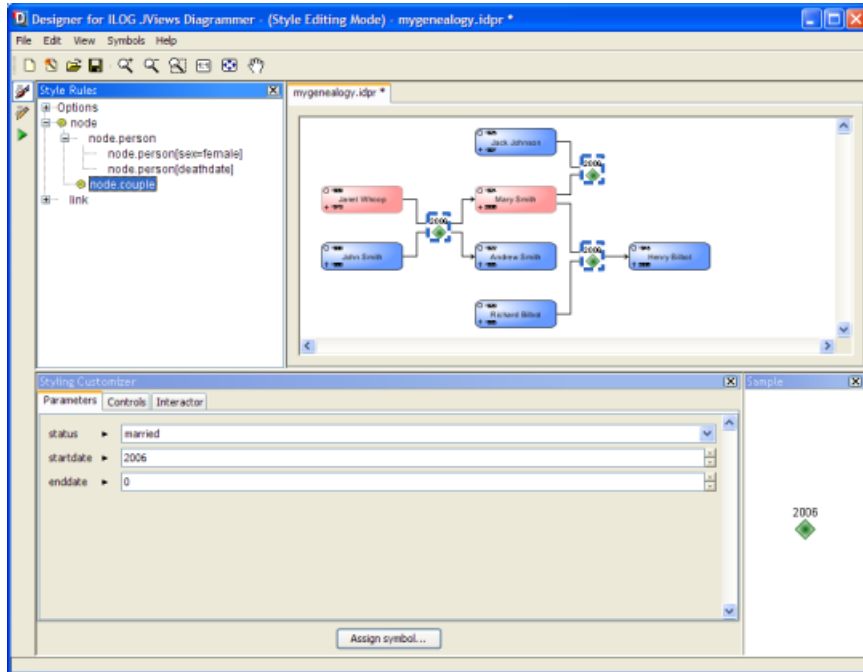
### To assign a different symbol to the couple nodes:

1. Select the new `node.couple` rule in the Style Rules pane.
2. Click Assign symbol at the bottom of the Styling Customizer.

The Assign Symbol dialog box displays.

3. Select the symbol Couple in the Genealogy category.
4. Click Apply.

See the result in the following figure.



*Assigning the couple symbol to nodes*

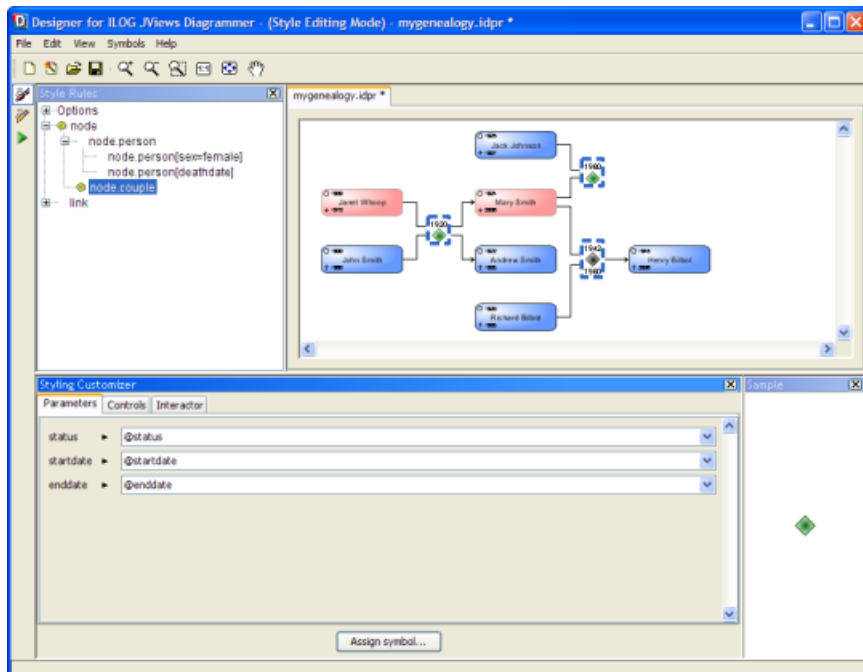
## Setting properties in the Styling Customizer for couple nodes

With the new `node.couple` rule selected, you will set the status and date properties in the Styling Customizer.

### To set properties for the couple nodes:

1. Select the `node.couple` rule in the Style Rules pane.
2. In the Parameters tab of the Styling Customizer, set the status to `@status`:
  - ◆ Right-click in the field.
  - ◆ Choose Enter an Expression.
  - ◆ Delete the current value.
  - ◆ Select `@status` from the list of available expressions.
  - ◆ Press ENTER.
3. Set the start date to `@startdate` and the end date to `@enddate` in the same way.
4. Press ENTER.

You can see the diagram change as you work, see the following figure.



*Customized couple nodes*

The Couple symbol has been designed to change colors according to the status: green for married couples, black for divorced couples.

You can see that `person` nodes and `couple` nodes now look quite different to represent the semantics of the data. You can see your new rules in the `node` section of the Style Rules.

---

## Rule-select and yellow icons

If you click an object in the diagram, the most specific rule that affects that object is selected in the tree and also has a yellow icon. This action is called *rule-select*. Less specific rules that are applicable to the object also have the yellow icon, see figures *Customized person nodes* and *Customized couple nodes*.

In the diagram, the selected object is surrounded by a selection box. Further objects to which the selected rule applies are also surrounded by selection boxes. Objects to which the rule applies, but for which more specific rules may override that rule's settings, are surrounded by thin selection boxes.

If you change the selected rule, the change is sure to affect all the objects shown with a (thick) selection box; it may or may not affect the objects shown with a thin selection box. Keep clicking on a given graphic object to switch between and select the rules marked with a yellow icon.

---

## Status icons and local properties

In the Styling Customizer, a right arrowhead indicates that the property has been set locally, that is, in the selected rule. An upward arrowhead indicates that the property value has been inherited from a previous rule.

# ***Customizing genealogy links by type***

Explains how to make a `couple` link look different from a `child` link.

## **In this section**

### **Creating a couplelink rule**

Explains how to create a rule for couple links.

### **Setting couplelink properties**

Explains how to change the styling of couple links.

---

## Creating a couplelink rule

### To create a couplelink rule:

1. Select the `link` rule in the Style Rules pane.
2. Choose **Edit>Create Style Rule**.  
The Create Style Rule Wizard opens.
3. Click Next.  
The Conditions page of the Create Style Rule Wizard displays.
4. Uncheck **Generated Name** and type `link.couplelink`.
5. Select `user-defined type` in the first drop-down list.
6. Type `couplelink` in the field next to the second drop-down list.
7. Click Finish.

The new rule appears in the Style Rules as `link.couplelink` and is selected.



## Setting couplelink properties

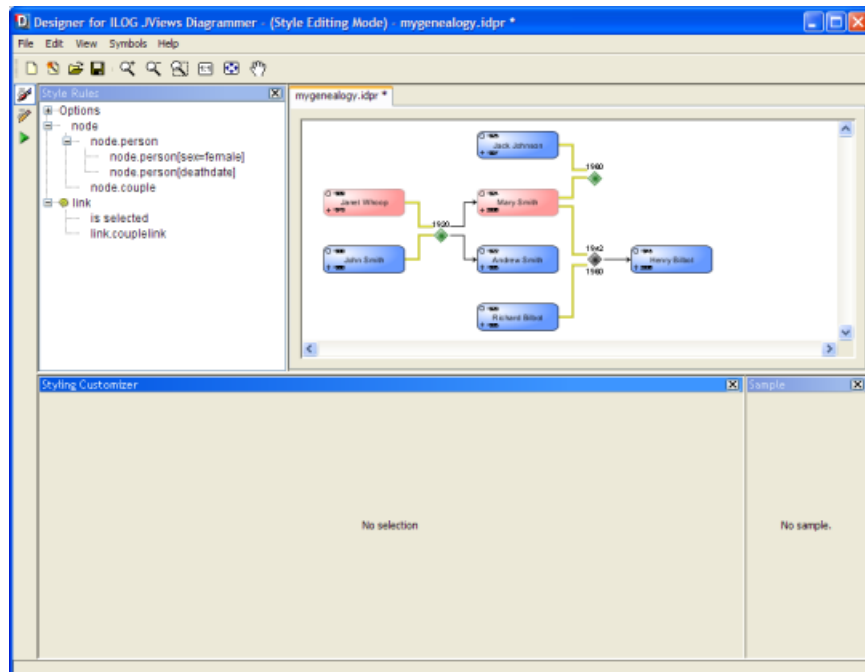
You will change the styling of couple links in the Styling Customizer.

### To set styling properties for couple links:

1. Make sure the `link.couplelink` rule is selected in the Style Rules pane.
2. In the Width tab of the Styling Customizer, increase the Line width to 2 and the Line width max to a value greater than 2.
3. In the Skin tab, set the Foreground color to yellow and change the Mode to give an outlined effect.
4. In the Arrow tab, change the Mode to remove the arrowhead.

**Tip:** It may be easier to see the result without selection boxes around the links: click in the canvas to deselect.

The result should be as shown in the following figure.



*Customized couple links*

**Tip:** Use the Fit to Contents button in the toolbar to resize the diagram so that you can see all of it.

---

## Testing application behavior

When you added facts to your person nodes, you set tooltips on these facts. These tooltips will appear in the final application. Therefore, you need to be in Preview Mode to see the tooltips.

By default, in the Designer you are in Style Editing Mode. In Style Editing Mode, you can define style rules as you have been doing.

### **To switch to Preview Mode:**


- ◆ Click the Preview button  in the vertical toolbar.

Preview Mode shows you the diagram as it will be in an application. This mode allows you to test the application behavior.

---

## Adding and customizing a legend

The legend is an explanatory table listing graphic symbols and explaining their meanings.

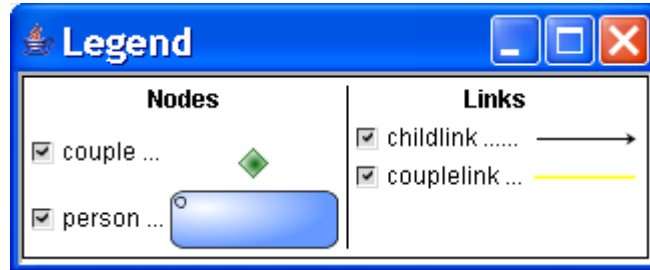
Switch back to Style Editing Mode by clicking the Style Editing button  in the vertical toolbar.

### To add a legend to your diagram:

1. Choose **Edit>Options**.
2. Check Legend.
3. Click OK.

The Legend option adds a legend to your diagram.

The Legend window can be inlaid within the diagram or can be an independent window, see the following figure.



*A separate legend window*

### To display the legend as an independent window:

- ◆ Uncheck the Inlaid option in the Position tab of the Styling Customizer.

By default, the legend shows two columns, one for nodes and one for links, with an entry for each user-defined type.

You can customize the general appearance, the position and the contents of the legend.

### To customize the general appearance of the legend:

1. Make sure Legend is selected under Options in the Style Rules pane.
2. In the Styling Customizer, use the Display tab to customize the font, spacing, colors of the legend.

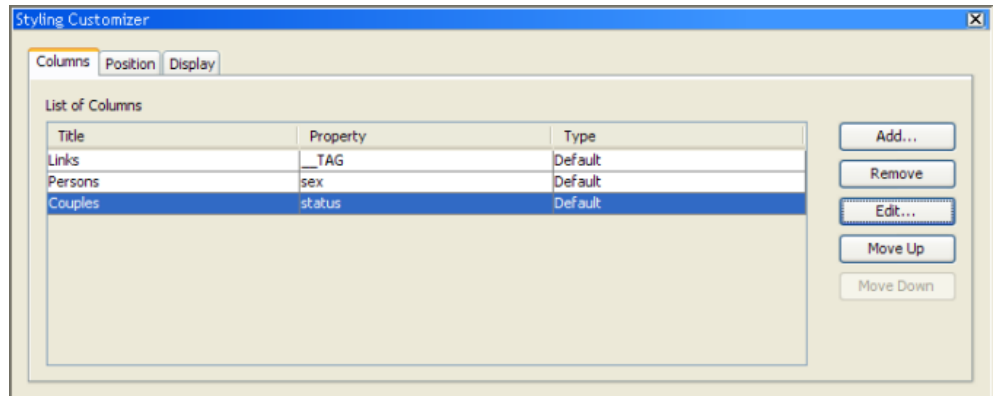
### To customize the position of the legend:

1. Make sure Legend is selected under Options in the Style Rules pane.
2. In the Styling Customizer, use the Position tab to customize the position of the legend.

### To customize the legend contents:

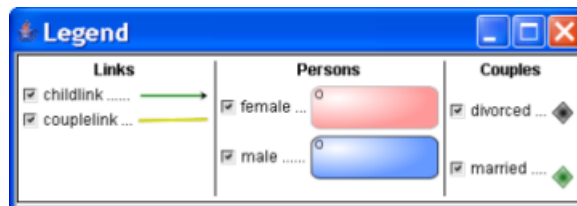
1. Make sure Legend is selected under Options in the Style Rules pane.

2. In the Styling Customizer, use the buttons in the Columns tab, see the following figure.



*Customizing a legend*

A possible customized legend is shown in the following figure.



*A customized legend*

In this customized legend, there are the following:

- ◆ A different title for the column `Nodes: Persons`, created with the Edit button in the Styling Customizer, and customized to use the `sex` property.
- ◆ An extra column for Couples, created with the Add button in the Styling Customizer, and customized as before to use the `status` property

### Filtering

The Legend provides a filtering function.

For example, in the Legend, clear a check box and you will see that the nodes or links of that type disappear.

---

## Integrating your development into an application

When you have saved your diagram in the Designer, you have a project file that you can use to integrate the diagram into an application.

You can integrate the complete diagram into an application through the project file, or you can integrate only the style sheet to have the styling developed in the Designer applied to real data in an application.

You need to add the symbol palettes used in the diagram to the classpath of the application, otherwise the symbols cannot be found and the project file cannot be executed.

### To integrate the genealogy project file into an application:

1. Create a new diagram component.
2. Load the project file.

See the following code sample which illustrates these two steps. It shows you how to integrate the genealogy example in Java™: first you create a new diagram component and then you load the project file into it.

#### Loading a project file into a diagram component

```
...
    IlvDiagrammer diagrammer = new IlvDiagrammer();
...
try {
    // Load a project file.
    //
    diagrammer.setDataFile(new URL("file:data/mygenealogy.idpr"));
} catch (Exception e) {
    e.printStackTrace();
}
...
```

For more information on writing applications that make use of diagram components, see *Writing an application*.

The Designer makes it much easier to write the style sheet. You can use the Designer for styling a sample of dummy data or an XML dump of your data. Then you can integrate the style sheet from the Designer into your application. You apply the styling defined in the style sheet to the real data used in the application.

### To integrate an existing style sheet into an application:

1. Make sure the project style sheet is accessible from the application.  
The `.css` file must be accessible from the application.
2. In the application, after creating the diagram and loading the data model, use the following code:

#### Styling the Diagram Component

```
try {
```

```
diagram.setStyleSheets(new String[]{"..."});  
} catch (IlvStylingException x) {  
    System.err.println("Cannot load style sheets: " + x.getMessage());  
}
```

See The style sheets in *Developing with the JViews Diagrammer SDK* for more information on how to load style sheets.

---

## Reusing a project in the Designer

You can easily refine an existing project in the Designer. You may want to refine a prototype or make changes to the styling. The Designer offers a quick and easy way of doing this.

You can only reuse projects in the Designer that have been developed exclusively with the Designer. Projects that have been further developed in the SDK are likely to contain features that are not available in the Designer.

You can drag a project file from a directory into a Designer window. The diagram configured by the project file opens automatically in the Designer. The drag-and-drop action is available in both Style Editing and Diagram Editing modes.



# ***Getting to know the Designer***

Describes the main features of the Designer and explains how to use them to create JViews Diagrammer diagram components in general terms.

## **In this section**

### **Style rules**

Describes the features related to the definition of style rules.

### **Graph and link layouts**

Introduces how to select and customize graph and link layouts.

### **Label layout**

Introduces the label layout and how to customize it.

### **Symbols**

Explains how to use symbols when designing a diagram.

### **General links**

Lists the options available in the Styling Customizer pane when customizing links.

### **Link connections**

Provides details about the ports on a symbol to which links connect.

### **Interactors**

Introduces the concept of interactors in JViews Diagrammer and explains how to customize them.

### **Working in different modes**

Explains how to use each of the three modes available in the Designer.

**Using wizards**

Describes the wizards provided with the Designer to guide you through the creation of diagrams and style rules.

**The Rule menu**

Describes the available commands of the Rule menu.

**The Undo and Redo facilities**

Lists the actions that you can undo and redo.

**The Check Rules facility**

Explains what the Check Rules command does.

**Messages for troubleshooting**

Describes the Messages pane.

**Printing the diagram**

Describes the print options.

**Comparing the Designer and the API**

Presents the pros and cons of the Designer and the API.

# *Style rules*

Describes the features related to the definition of style rules.

## **In this section**

### **Style Rules pane**

Presents the content of the Style Rules pane

### **Options**

Lists the options available in the Style Rules pane of the Designer window.

### **Node and link rules**

Describes what a data rule (node or link) consists of.

### **Level of detail**

Describes the three levels of detail that you can use to display a diagram.

### **Styling customizers**

Lists the interface elements of the Styling Customizer pane and their purpose.

### **Styling properties**

Describes the Styling Properties pane of the Designer which lists the properties associated with the nodes and links of a diagram.

### **Priority**

Explains the order of priority in which style rules apply.

### **Style sheet debugging**

Provides hints to understand why a change might not be applied.

### **Resetting of default property values**

Explains that you can revert to the default values of properties.

---

## Style Rules pane

The Style Rules pane displays the options for styling your diagram as a whole and data rules for styling the application data. When you save a diagram as a project, the style rules (options and data rules) are saved in a style sheet (.css file) with the CSS for Java syntax. For details of the syntax, see Using CSS syntax in the style sheet in *Developing with the JViews Diagrammer SDK*.

---

## Options

The Options section of the Style Rules pane allows you to enable or disable various facilities at the level of the diagram as a whole. Each diagram you work on has its own style sheet and therefore its own options.

There are options for the following aspects of the diagram:

◆ Background

The Background option allows you to display a color, a pattern, or a graphic in the background of your diagram.

◆ Background Map

The Background Map option allows you to add a background map and to position objects in a georeferenced way (with latitude and longitude). For an example, see *Using background maps*. For full details of map styling, see The Map renderer in *Developing with the JViews Diagrammer SDK*.

◆ Blinking

The Blinking option allows you to add blinking to the symbols that make use of this feature. You can define the blinking period. For full details of blinking styling, see The Blinking renderer in *Developing with the JViews Diagrammer SDK*.

◆ Graph Layout

The Graph Layout option handles the relative positioning of the data objects in the diagram. See also *Graph and link layouts*. For an example, see *Using the Graph Layout Wizard*. For full details of graph layout styling, see The GraphLayout renderer in *Developing with the JViews Diagrammer SDK*.

◆ Info Balloons

Info Balloons are text elements attached to a node or link but displayed only when the object is clicked. For full details of Info Balloon styling, see The InfoBalloon renderer in *Developing with the JViews Diagrammer SDK*.

◆ Legend

The Legend option activates an optional legend displayed by default at the lower left of the diagram. The legend provides an explanatory function and a filtering function. For an example, see *Adding and customizing a legend*. For full details of legend styling, see The Legend renderer in *Developing with the JViews Diagrammer SDK*.

◆ Label Layout

The Label Layout option allows you to place the labels of the nodes and links so that they do not overlap with the nodes and links.

◆ Link Layout

The Link Layout option handles the positioning of links to avoid them overlapping with one another or with nodes. For full details of link layout styling, see The LinkLayout renderer in *Developing with the JViews Diagrammer SDK*.

◆ Subgraphs

The Subgraphs option allows you to expand a node by clicking the handle in the corner to display subnodes within it. This option is compatible with XML data only. For full details of subgraph styling, see The SubGraph renderer in *Developing with the JViews Diagrammer SDK*.

◆ Zoom Thresholds

Zoom thresholds prevent the node size from becoming too small or too large when you zoom. They are generally most useful when you have a background map because the map size and the node sizes may be very different.

There is a Styling Customizer for each of these facilities to set the styling properties. The Styling Customizer appears in the lower pane of the Designer when you select the option in the Style rules pane.

---

## Node and link rules

A node or link rule (data rule) consists of an object type and attribute conditions followed by declarations.

A data rule is defined on a group of data model objects belonging to the same class, or having a common ascendant class, or belonging to the same user-defined type. It is applied to the graphic objects used to visualize the data model objects in the diagram.

In the Designer, you see the data rules (node and link) in the top left pane, titled Style Rules, following the Options. This pane shows a hierarchical view of the rules depending on how specific they are.

In the Designer, you see the declarations in terms of property values in the Styling Customizer or the optional list of Styling Properties.



---

## Level of detail

The style rules allow you to define three different levels of detail for displaying the nodes and links in a diagram: high (default), medium, low.

For example, you could define the following information to be displayed for each detail level:

- ◆ High level: the nodes have a fill and a label
- ◆ Medium level: the nodes have no fill but they have a label
- ◆ Low level: the nodes have no fill and no label

A low level of detail allows you to speed the rendering of the diagram.

Once these levels have been defined through the Create Style Rule Wizard and parameterized through the Styling Customizer, you can switch between them in two different ways:

- ◆ By selecting a different level of detail rule in the Style Rules pane
- ◆ By choosing a different level of detail in the **View>Level of detail** submenu.

Only rules that match the selected level of detail contribute to the customization of the diagram.

A rule that does not define any level of detail matches all levels of detail.

The level of detail can also be specified through the API with the `setDetailLevel(java.lang.String)` method.

---

## Styling customizers

The Styling Customizer is a user-friendly way to see properties and their values, subdivided into categories by different tabs and with explanatory names. You use a Styling Customizer to customize an option or a set of graphic objects by setting property values. The Styling Customizer for the current rule selection is displayed in the lower pane of the Designer.

For each property, the Styling Customizer displays one of the following according to the type of value concerned:

- ◆ A check box to select or clear
- ◆ A list of the valid values or items for you to choose from
- ◆ A step scale for numerical values
- ◆ A text field for you to enter text
- ◆ Buttons for you to add or remove values or lines of text
- ◆ An Ellipsis button to browse in order to select a file name or font
- ◆ A choice of the following systems for color settings:
  - JViews color disk
  - RGB (Red, Green Blue)
  - HSB (Hue, Saturation, Brilliance)
  - Swatches

---

## Styling properties

The Styling Properties list gives a view of properties that is closer to the API; it comprises a single exhaustive list and shows the API (short) names. To open the Styling Properties pane, choose **View>Styling Properties**.

The list shows the locally set properties (those set in the selected rule), in alphabetical order, followed by the inherited properties (those set in previous rules), in alphabetical order. The status icons next to each property show which property values are local and which are inherited:

- ◆ Arrow up : value set in a previous rule
- ◆ Arrow right : value set in the selected rule

Clicking the arrow icon toggles the status.

**Warning:** If you change the status to Inherited from a previous rule, you will lose the value set by the selected rule.

---

## Priority

The principles of inheritance and overriding apply to style rules. This allows you to write default rules for all model objects of a certain type and then to refine the customization with more specific rules that apply to subsets of these model objects.

The priority of a style rule depends on how specific it is and on the order with respect to other style rules of the same specificity. In general, a rule with more components in its selector has a higher priority because it is more specific. Note that a more specific rule sets property values for some model objects but not for others. When the specificity of two rules is the same, the lowest rule in the tree has priority.

Different rules apply to different groups of objects, but several rules can apply to the same object. If the same property is set on one object in several rules, only the setting in the rule with the highest priority is retained; this rule *overrides* all the previous ones.

To style a model object, the Designer sorts all rules that match it in priority order, and then applies the settings they contain to the graphic object representing the model object.

---

## Style sheet debugging

If you change a property in a Styling Customizer, and you do not immediately see the effect of the change, this is probably because the selected rule has been overridden by a later rule. You can look at the other rules at the same level in the tree to see if the same property is set in one of these rules.

---

## Resetting of default property values

A graphic object can have default values for its properties, which can be set at creation time.

In a style sheet, some property values may be inherited at all levels and therefore come from the graphic object's default values. If you change such a property value in a style rule to reflect a temporary state (such as selection), when the temporary state no longer applies (the object is deselected), the property value does not automatically revert to the default value. This situation arises because there is no style rule to set the default value. You therefore need to include this setting explicitly in the general rule for the node or link. For an example, see *Resetting a default value after a state change*.

If you do not change such a property value according to a temporary state in a style rule, you can leave the implicit default value and not worry about it.

---

## Graph and link layouts

In the Designer, you can set the Graph Layout and Link Layout options in the Style Rules pane, in order to lay out the nodes and links in your graph according to a specific arrangement.

The graph layouts available are: bus, circular, grid, hierarchical (the default), tree, uniform length edges.

The link layouts available are: hierarchical, short links, long links.

---

### Layout selection

There are many ways to select a layout, and you can choose one according to which is most convenient for you:

- ◆ Choose a layout theme in the Graph Layout Theme page of the New Diagram Wizard or in the Graph Layout Wizard (see also *Graph Layout wizard*). Each layout theme is a ready-customized layout for a specific purpose.
- ◆ In the Styling Customizer for the Graph Layout option or the Link Layout option, select a layout type from the vertical toolbar, or in the Control tab.

The most commonly used layout is hierarchical. An example project with the hierarchical layout is the **genealogy example**, `<installdir>/jviews-diagrammer86/bin/designer/data/examples/genealogy.idpr`. Example projects with bus and circular layouts are also supplied in `<installdir>/jviews-diagrammer86/bin/designer/data/examples`.

---

### Layout customization

If you need to customize the default behavior of a graph or link layout algorithm, or the parameters set by a layout theme, you can select Graph Layout or Link Layout under Options and set properties in the Styling Customizer.

The most common properties to use are those in the Control and Basic tabs.

Where necessary, online Help text is available for individual properties.

A few graph and link layout properties are set on a per-node or per-link basis. Explanations and instructions for setting these appear in the Styling Customizer. For an example, see *Setting a link style with mixed links*.

---

## Label layout

In the Designer, you can set the Label Layout option in the Style Rules pane, in order to lay out the labels of nodes and links so that they do not overlap and to display the labels outside the nodes. You can specify label layout for either nodes or links, or for both.

If you open a project with a style sheet that enables label layout in the Designer, the labels of nodes and/or links are laid out as soon as the style sheet is loaded.

Please note that the labels of nodes and links have a position which you can customize in the Styling Customizer for nodes and links respectively. When you enable the Label Layout option, it takes precedence over the positioning of the labels specified in the nodes and links styling customizers.

---

### Label layout customization

If you need to customize the default behavior of the label layout algorithm, you can select Label Layout under Options and set properties in the Styling Customizer.

The most common properties to use are those in the Control and Geometry tabs.



# ***Symbols***

Explains how to use symbols when designing a diagram.

## **In this section**

### **About symbols**

Explains what a symbol is in the context of the Designer.

### **Styling Customizer**

Describes the content of the Styling Customizer pane when a symbol is selected in the diagram.

### **Assigning a symbol**

Explains how to apply a specific symbol to nodes.

### **Editing a symbol**

Explains how to access the Symbol Editor from within the Designer.

### **Refreshing symbols**

Explains how to update the view when the symbols have changed.

### **Palette management**

Explains the different ways to manage palettes of symbols within the Designer.

---

## About symbols

In the Designer, you use symbols to represent the nodes in the diagram. A symbol is a graphic object made up of several elements. Symbols can be created and edited through the Symbol Editor which is accessible through the menu **Symbols>Edit symbols** of the Designer.

---

## Styling Customizer

When a symbol is selected in the diagram, the Styling Customizer displays three tabs: Parameters, Controls, and Interactor.

- ◆ **Parameters:** Each symbol exports specific parameters that customize its graphic representation. These parameters can be graphic properties (such as color, font) or state information (such as sex, birthdate).
- ◆ **Controls:** These are graphic properties that are shared by all the symbols and apply to the symbol globally (tooltip text, visibility...).
- ◆ **Interactor:** This is a graphic property that is shared by all the symbols regarding user interaction (Enable interaction). This tab also lists the symbol parameters that are modified when user interaction is enabled.

The parameters are customizable through the Symbol Editor when the symbol is selected.

## Assigning a symbol

You can replace the symbol assigned to a specific rule.

### To replace the symbol:

1. Select the symbol in the diagram.

-- or --

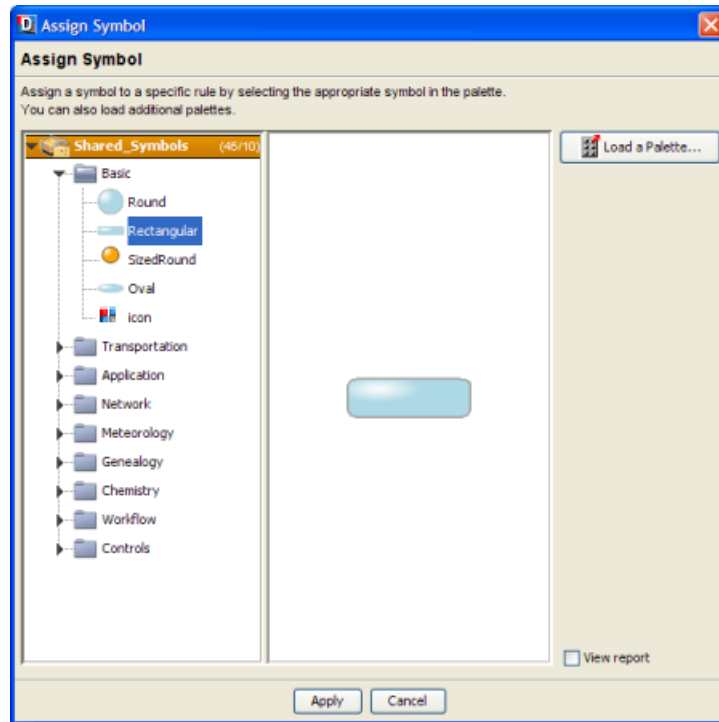
Select the corresponding rule in the Style Rules tree.

2. Choose **Symbols>Assign symbol** in the menu bar.

-- or --

Click the Assign symbol button in the Styling Customizer.

This displays the following dialog box:



*The Assign Symbol dialog box*

3. Click the button Load a Palette to select a palette from which to choose a symbol.

-- or --

Select a symbol in one of the palettes available in the left pane.

The right pane shows a preview of the selected symbol.

4. Check the View report option if you want to display a report on the selected symbol.  
This report includes the symbol description and parameter details.

---

## Editing a symbol

You can edit symbols in the Symbol Editor which is available from the Designer.

### To open the Symbol Editor:

- ◆ Choose **Symbols>Edit symbols** in the menu bar.

If no symbol was selected in the diagram, the Symbol Editor opens on an empty window.  
If a symbol was selected, the Symbol Editor opens on this symbol and allows you to customize it.

---

## Refreshing symbols

You can rebuild all the symbols if they have changed in the palette.

- ◆ Choose **Symbols>Refresh symbols** in the Designer.

---

## Palette management

The command **Symbols>Manage palettes** displays the list of all the palettes used in the diagram, with their location.

You can:

- ◆ Load new palettes
- ◆ Close existing palettes (only if they are not used)
- ◆ Preview a symbol when you select it
- ◆ View a report (details) about a palette or a symbol

---

### Palette lookup

The palettes need to be added to the classpath of the application before deployment. However, the Designer requires extra information to correctly manage the symbol customizers. For external palettes, this information is retrieved only if the Designer knows the absolute path of the palettes. In this case, the palettes do not need to be in the Designer classpath.

During a Designer session, an external palette is loaded directly from the Assign Symbol dialog box, through a file selector. The absolute file path of the palette is then known and saved in the style sheet when the project is saved. If the project is loaded in another session, make sure the external palette path has not changed, otherwise the Designer has trouble working out the symbols and a warning message may be displayed.

To ease development, a fallback rule tries to load missing palettes from the project directory (the directory containing the `.idpr` file).

Please note that, as the Designer environment is very particular, the external palettes must be in the classpath of all deployed applications.



---

## General links

In the Designer, an `IlvGeneralLink` is used to represent the links between the nodes in the diagram. You can customize the links in the Styling Customizer through a set of tabs:

- ◆ Shape: Choose the end, the join, the crossing mode of links
- ◆ Width: Specify the width of the link, as well as minimum and maximum widths
- ◆ Skin: Specify a gradient or a texture effect
- ◆ Arrow: Specify the type of arrow or no arrow, as well as the color of the arrow
- ◆ Border: Specify the width of the border, a color for the upper edge of the border and a different color for the lower edge
- ◆ Label: Place a text label at a given distance from the link
- ◆ Tooltip: Specify a tooltip for the link
- ◆ Ports: Configure the ports to which the links are attached on a symbol

You can also customize these aspects through CSS. Please refer to Customizing general links in the style sheet in *Developing with the JViews Diagrammer SDK*.

For details about the ports configuration, please refer to The Composite renderer in *Developing with the JViews Diagrammer SDK*.

---

## Link connections

The Ports tab in the Styling Customizer for links allows you to define the origin of a link as one of the OUT ports of a symbol. Similarly, you can define the destination of a link as one of the IN ports of a symbol. Ports on symbols are defined using the Symbol Editor (see Using link-in and link-out ports in the *Using the Symbol Editor* user documentation).

The Controls tab in the Styling Customizer for nodes allows you to define the link connector mode. Available modes are: FREE, CLIP, and CENTER. The link connector mode controls the connection of all the links that are incident to a particular node. To specify a link connector for each link individually, use the Ports tab. For details about link connectors, see Contact points in the *Essential JViews Framework* user documentation.

Please note that, when graph layout is enabled, the link connection points may be modified by the layout algorithm used (see Inappropriate-link exception in the *Using Graph Layout Algorithms* user documentation).

The configuration of link connections is applied in the following order of priority:

1. The graph layout configuration (if graph layout is enabled)
2. The configuration defined in the Ports tab of the link Styling Customizer
3. The configuration defined in the Controls tab of the node Styling Customizer
4. The FREE connector mode

# *Interactors*

Introduces the concept of interactors in JViews Diagrammer and explains how to customize them.

## **In this section**

### **About interactors**

Shortly explains what interactors are.

### **Predefined interactors**

Lists the interactors that are provided in the Designer.

### **Customizing interactors**

Describes the ways to customize interactors in the Designer.

---

## About interactors

An interactor handles the behavior of the whole display or of an object within it. Predefined interactors are active in the Designer and provide the conventional behavior required for display, navigation, monitoring and editing. These same interactors can be included in your application: they are supplied as JavaBeans™ that are GUI components.

---

## Predefined interactors

The following predefined interactors are active in the Designer for JViews Diagrammer:

- ◆ Pan, in all modes
- ◆ Select, in all modes
- ◆ Zoom, in all modes
- ◆ Display an info balloon (with the Info Balloon option only)
- ◆ Expand or collapse nodes (with the Subgraphs option only)
- ◆ Make link (in Diagram Editing mode only)
- ◆ Make node (in Diagram Editing mode only)
- ◆ Edit label (in Diagram Editing mode only)



# *Customizing interactors*

Describes the ways to customize interactors in the Designer.

## **In this section**

### **Customization options**

Lists the options to customize interactors.

### **Linking predefined interactors to parameters**

Explains how to link a predefined interactor to a parameter of the diagram.

### **Controlling interactions using CSS functions**

Explains how to obtain a finer control of symbol interactions.

### **Customizing the diagram dynamically**

Explains how to define dynamic interactions.

---

## Customization options

To implement customized behavior in your application, do one of the following:

- ◆ Use the Designer to connect a parameter to a predefined interactor
- ◆ Use the Java™ API to add a custom interactor

This section will concentrate on linking interactors to style rules. For information on how to create customized interactors using the Java API, see [Connecting interactors to diagrams using listeners](#) in *Developing with the JViews Diagrammer SDK*.



## Linking predefined interactors to parameters

Each symbol has a set of built-in interactors that can be utilized to make symbols dynamic. For more information, see Understanding interactors in the *Using the Symbol Editor* user documentation. Using the Designer, you can connect the string value of the `push_state` internal parameter to a custom data property. You then use style rules to make interactive changes to your diagram as the data property changes value. For example, you can change the symbol used to represent a node when it is clicked.

**To change the symbol used to represent a node when it is clicked:**

1. Open the Designer.
2. Create a new default document by selecting **File>New**.
3. Double-click Node 1.
4. Click the New Properties icon in the Data Properties pane.

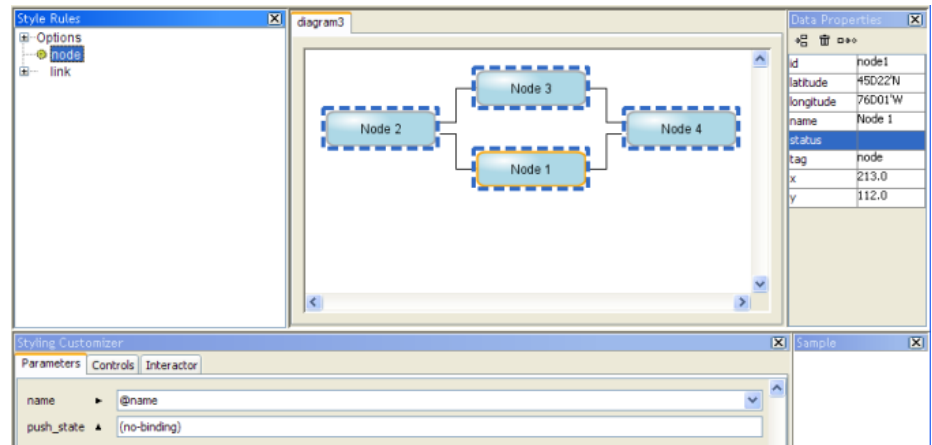


*The New Properties Icon*

The Property Name window appears.

5. Type `status` in the Enter the name of the new property field.
6. Click OK.

The status property is now visible in the Data Properties pane.

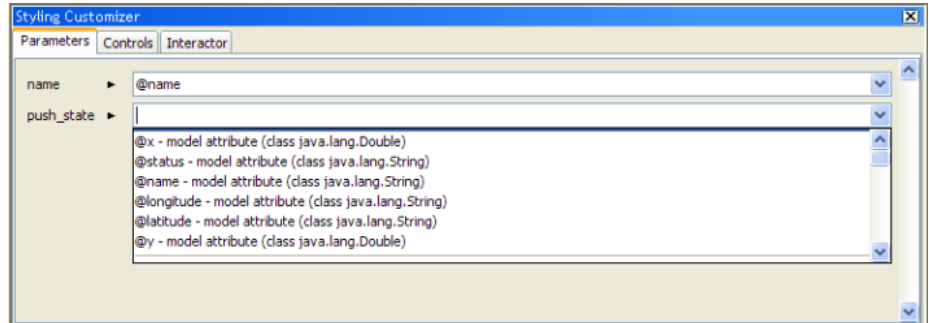


*Selecting a node in a default diagram*

7. Right-click the `push_state` field in the Parameters tab of the Styling Customizer. A shortcut menu appears.
8. Select Enter an Expression in the shortcut menu.

9. Delete any text in the `push_state` field.

A list of available properties and functions appears.



*Selecting a string parameter to capture the `push_state`*

10. Select the `@status` property.

The `status` property has been created to hold the state of the `push_state` parameter.

11. Run your diagram in Preview mode and click Node 1, then return to Style Editing mode.

You see that the value of the `status` property has changed to `BUTTON_CLICKED` in the Data Properties pane.

#### **To change the current mapping:**

1. Go to the Parameters tab.
2. Edit the parameter value.

---

## Controlling interactions using CSS functions

The CSS functions `stateInteractor()` and `numericInteractor()` allow you to handle symbol interactions in a finer way.

These functions have two optional parameters. The first parameter is the name of a model attribute that will hold the interaction values both in read and write modes. The second parameter is the initial value, in case the model attribute does not yet exist.

If the first parameter is not specified, then an internal variable is used (actually a property of the graphic object). This can be useful if you do not want the model to be affected by interactions.

These functions are able to call an `ActionListener` when interaction is performed. To do so, use `addActionListener()` on the function itself. See `addActionListener`.

### Examples

In the Parameter tab, set the following values to the interactor parameter (like `push_state`):

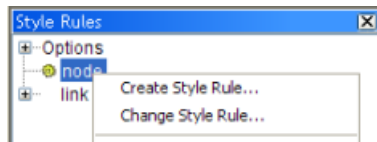
- ◆ `numericInteractor(test, 40)`: interaction value for a dial or a slider, of type numeric, that modifies the model attribute `test` and defaults to 40 if the model has no value defined for `test`.
- ◆ `numericInteractor(, 80)`: interaction value for a dial or a slider, of type numeric, that does not affect the model and defaults to 80.
- ◆ `stateInteractor(state)`: interaction value for a button, of type string, that modifies the model attribute `state`. The default value is given by `state`.

## Customizing the diagram dynamically

Style Rules change the aspect of a node or link when one of the data properties attains a specific value. In *Linking predefined interactors to parameters*, you connected the state of a button to the @status data property.

**To add dynamic customization to a diagram using the Designer, you will use a style rule to change the aspect of a node as the @status property changes, that is, when a user clicks a node:**

1. Open the diagram you created in *Linking predefined interactors to parameters*.
2. Right-click `node` in the Style Rules pane and select `Create Style Rule` in the shortcut menu.

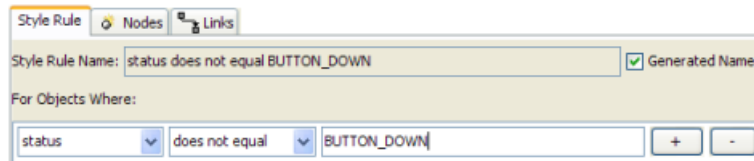


*Create Style Rule command*

The Create Style Rule Wizard appears.

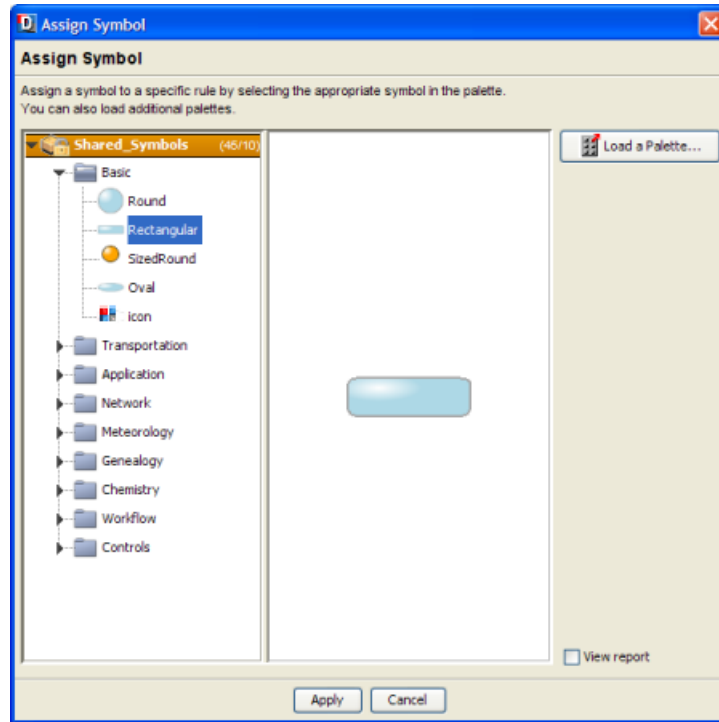
3. In the Create Style Rule Wizard, click `Next`.
4. In the Style Rule tab, set the following condition:

`status does not equal BUTTON_DOWN`



*Creating a new style rule*

5. Click `Finish`.
6. Click the `Assign` symbol button in the Styling Customizer.  
The Assign Symbol dialog opens.
7. Select the `Rectangular` symbol.



*Assigning the rectangular symbol to the style rule*

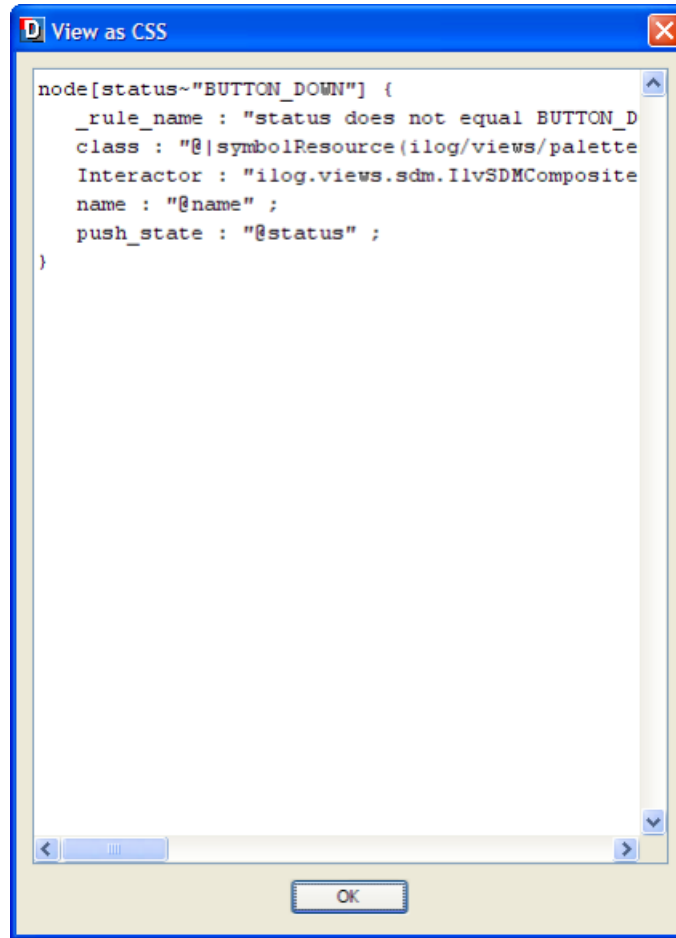
8. Click Apply.
9. Select the `status does not equal BUTTON_DOWN` rule in the Style Rules pane.
10. Set the value of the `push_state` field in the Styling Customizer to `@status` for this rule.

You have created a style rule that ensures the normal node symbol is displayed when a node is not being pressed. To verify that this rule has been correctly applied, validate the CSS.

**To validate the CSS:**

1. Right-click the `status does not equal BUTTON_DOWN` rule and select View in the shortcut menu.

The View as CSS dialog appears.



*Viewing the CSS of a style rule*

2. Once you have validated the CSS rule, click OK.

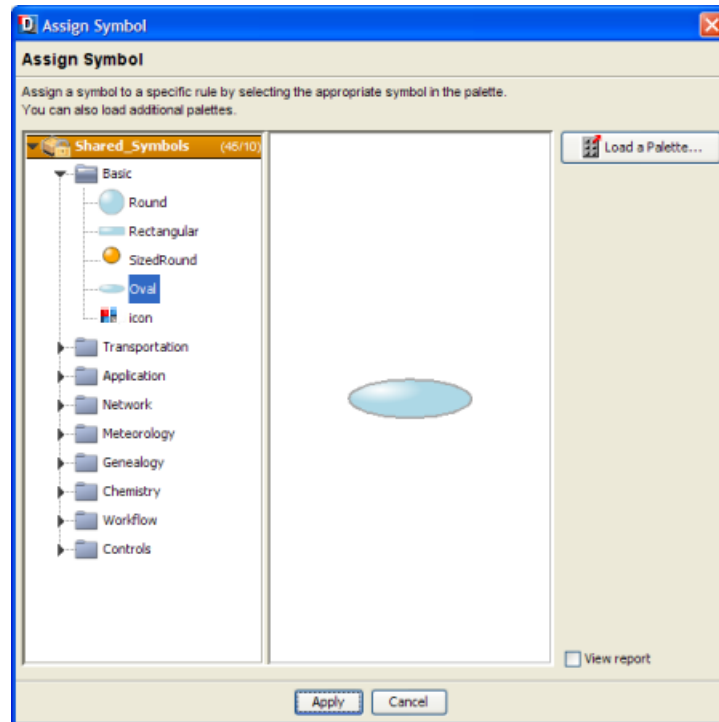
**To display a different symbol when a node is pressed down:**

1. Right-click `node` in the Style Rules pane and select Create Style Rule in the shortcut menu.

The Create Style Rule Wizard appears.

2. Click Next.
3. In the Style Rule tab, select `status equals BUTTON_DOWN`.
4. Click Finish.
5. Select the `status equals BUTTON_DOWN` rule in the Style Rules pane.
6. Click the Assign symbol button in the Styling Customizer.

The Assign Symbol dialog opens.



*Assigning the Oval symbol to the style rule*

7. Select the Oval symbol.
8. Click Apply.
9. Set the value of the `push_state` field in the Styling Customizer to `@status` for this rule.

The symbol used to represent a node now changes when any of the nodes in the diagram is clicked.





# *Working in different modes*

Explains how to use each of the three modes available in the Designer.

## **In this section**

### **The three modes**

Introduces the three modes.

### **Working in Style Editing mode**

Explains how to work in Style Editing mode.

### **Working in Preview mode**

Explains how to work in Preview mode.

### **Working in Diagram Editing mode**

Explains how to work in Diagram Editing mode.

---

## The three modes

Use the modes in the Designer to develop your diagram as follows:

- ◆ Work in Style Editing mode to define your style rules
- ◆ Work in Preview mode to test the resulting behavior of the diagram component as it would be in an application.
- ◆ Work in Diagram Editing mode to build up a diagram one node or link at a time, and to position nodes and links in a diagram manually by dragging them. Note that editing the diagram will modify the data model.

---

## Working in Style Editing mode

In Style Editing mode, you can define styling as follows:

- ◆ Set options to change some aspect of the diagram as a whole
- ◆ Create or change a style rule
- ◆ Select graphic objects and edit the associated style rules

### To enter Style Editing mode:

- ◆ Click the Style Editing button in the vertical toolbar.

### Panes in Style Editing mode

In Style Editing mode, the panes of the Designer window are as follows:

- ◆ Top left: Style Rules tree, showing the rules that determine the appearance of the objects displayed
- ◆ Right: Diagram in Style Editing mode
- ◆ Lower left: Styling Customizer for the selected rule, showing the styling property values.

You can display the data model as a reference if you open the Data Model window.

You can edit the data properties if you open the Data Properties window.

You cannot see all application behavior.

### To define a style rule:

- ◆ Choose **Edit>Create Style Rule** to create a new rule with the Create Style Rule wizard

-- or --

- ◆ Choose **Edit>Change Style Rule** to change the selected rule with the Change Style Rule Wizard.

You can also right-click a rule and choose Create Style Rule or Change Style Rule in the shortcut menu.

### To select a graphic object in the diagram:

- ◆ Click the object in the Diagram pane.

A rectangular dashed line appears around it. This is called the selection box.

### To deselect objects:

- ◆ Click in the pane, outside the objects.

If you select a rule in the Rule Tree, the rule is highlighted and all objects affected by the rule are selected in the diagram.

When a rule is highlighted, the Designer displays the Styling Customizer for the selected object(s).

Selection drives the property setting process as follows:

- ◆ When you select an option in the Style Rules, the Styling Customizer displays the associated properties.
- ◆ There can be several rules controlling a data object. When you select an object in the diagram, the Designer highlights the first rule for this object in the Style Rules. The first rule is the most specific one. This action is known as *rule-select*. If you click the same object again, the Designer highlights the second rule (second most specific) and so on, that is, it cycles through the rules from the most specific to the least specific and then back to the most specific again. At the same time, the Designer displays the associated properties in the Styling Customizer.

If you double-click a graphic object in the diagram, the objects properties are displayed in the Data Properties window. You can change the values of the properties, except the `tag` property (user-defined type).

**To change a property value:**

1. Click the name of the property.
2. Edit the value field of the property.

---

## Working in Preview mode

Preview mode allows you to see application behavior that you do not see in other modes, for example, it shows tooltips and Info Balloons if you have set any.

In Preview mode, you cannot change the style rules or the data model.

- ◆ Click the Preview button in the vertical toolbar to enter Preview mode.

### **Panes in Preview mode**

In Preview mode, a single pane is displayed containing your diagram.

---

## Working in Diagram Editing mode

In Diagram Editing mode you can insert nodes and links into a diagram and then position them. You can drag objects to new positions and change the way they are interconnected. When you update the diagram manually in this mode, the data model is automatically updated to remain synchronized with your diagram. For example, the data model changes to reflect new positions if you have a Map Diagram or an X/Y Diagram; the data model changes to reflect differences in interconnections.

In Diagram Editing mode, you cannot change the style rules nor see all application behavior.

### To enter Diagram Editing mode:

- ◆ Click the Diagram Editing button in the vertical toolbar.

### Panes in Diagram Editing mode

In Diagram Editing mode, the panes of the Designer window are as follows:

- ◆ Top left: Data Model tree, showing the model objects displayed
- ◆ Right: Diagram in Diagram Editing mode  
You can work using drag and drop in the Diagram.
- ◆ Lower left: Data Properties of the selected object, showing the current values.

### Selection in Diagram Editing mode

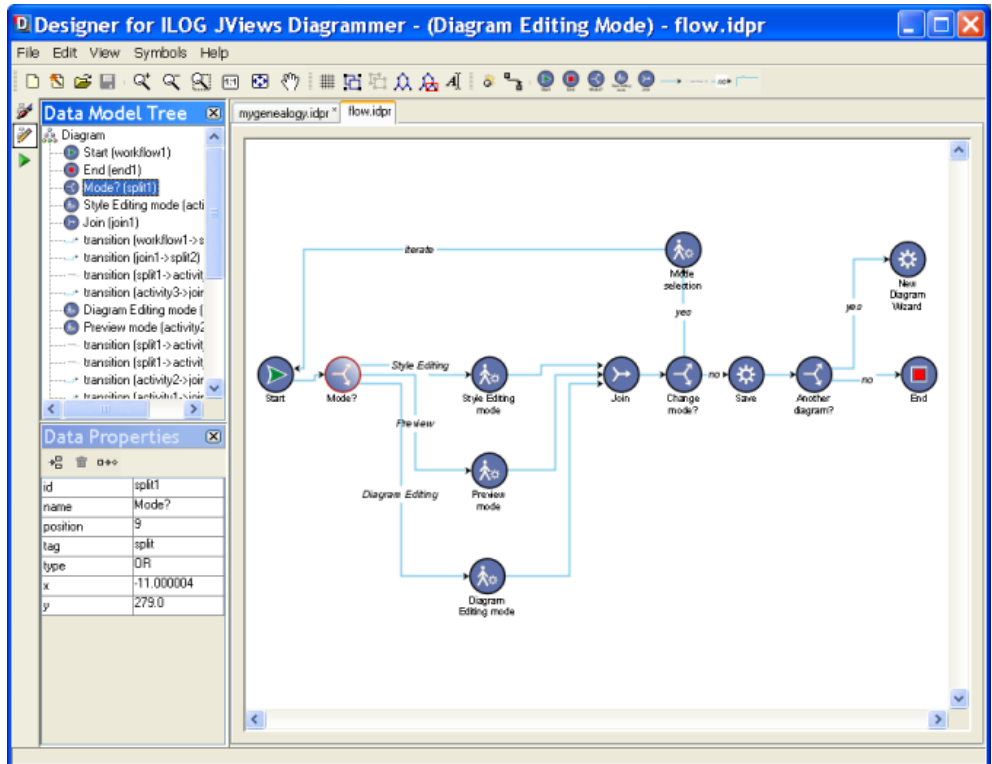
If you select objects in the tree or the diagram, they are displayed in the diagram according to the style rules you have defined that govern selection (if any). For example, if you have specified in a rule that a certain type of object should change color when selected by a user of your application, you will see this result when you select an object of that type. If you right-click an object, you can switch the selected/deselected state of the object.

### Diagram and data model modification

You can select a type of node from the right end of the horizontal toolbar and drop it onto the diagram. You can select a type of link and click its source node and destination node to create a link. Effectively, you add objects to your data model when you drop them onto the diagram.

Some data models are not editable, for example, data models derived from JDBC data sources. You cannot edit the diagram in this case.


The following figure shows a flow diagram. This example is available as project file `flow.idpr`.



The flow diagram developed in Diagram Editing mode


### To add a node:

1. Click the node icon in the horizontal toolbar.
2. Click anywhere in the diagram.

For example, to add a manual activity, click the manual activity icon  in the toolbar and then click below the existing manual activities in the flow diagram.

In addition to the nodes available from the toolbar, you can create new types of node.

### To create a new type of node:

1. Click the icon  in the toolbar.
2. Enter the type of the new node in the dialog.
3. Click OK.

The style of the new node is defined by the type set for the node. If no rule has been set for this node type, then the rule "node" applies alone.

**To move a node:**

1. Click the node and drag it to a new position.  
Any links attached to the node remain attached.
2. If it is useful, you can perform a layout on all the nodes or a selected subset of them, according to the layout algorithm currently configured in the style sheet.


**To add a link:**

1. Click the link icon in the horizontal toolbar.
2. Click one node in the diagram (source node).
3. Click another node (destination node).

For example, in the flow diagram, click the good link icon in the toolbar, then click the Mode? node and then click the new manual activity node.

In addition to the links available from the toolbar, you can create new types of link.

**To create a new type of link:**

1. Click the icon  in the toolbar.
2. Enter the type of the new link in the dialog.
3. Click OK.

The style of the new link is defined by the type set for the link. If no rule has been set for this link type, then the rule “link” applies alone.

**To move a link:**

- ◆ Click the end you want to detach and drag it to a different node.

The other end of the link remains attached as before.

If you perform a layout on all the nodes or a selected subset of them, affected links will change their routes.

**To add a user-defined type:**

1. Select a node.
2. Click the Set User-Defined Type button in the Data Properties toolbar.
3. Enter new values for the properties of the user-defined type.

Note that the `tag` property remains fixed and, instead, your user-defined type is identified by the value of the `CSSclass` property.

If you switch back to Style Editing mode and create a style rule specifying the appropriate `tag` value and `CSSclass` value, you can select a different icon in the Styling Customizer to represent the new user-defined type.

When you add a user-defined type, its icon is added to the horizontal toolbar in Diagram Editing mode automatically.



**To add a new property to a type of node or link:**

1. Click the New Property button in the Data Properties toolbar.
2. Enter the name and value of the new property.

**To remove a property of a type of node or link:**

1. Select the property in the Data Properties pane.
2. Click the Delete Property button in the Data Properties toolbar.



# *Using wizards*

Describes the wizards provided with the Designer to guide you through the creation of diagrams and style rules.

## **In this section**

### **New Diagram wizard**

Describes the wizard used to create a new diagram.

### **Create Style Rule wizard**

Describes the wizard used to create a style rule.

### **Change Style Rule wizard**

Describes the wizard used to change a style rule.

### **Graph Layout wizard**

Describes the wizard used to select a graph layout theme.

### **Data wizard**

Describes the wizard used to edit the data model.

---

## New Diagram wizard

The New Diagram Wizard allows you to specify the data on which a diagram is to be based and the initial look of the diagram. You can then refine the styling by working in Style Editing mode, see *Working in Style Editing mode*.

To open the New Diagram Wizard, choose **File>New from Wizard**. By default, this wizard is open when you run the Designer; you can disable this behavior from the first page of the wizard.

The New Diagram Wizard has the following pages:

- ◆ Welcome page.
- ◆ Choose the Format of Data to Import, with a list of the data formats.
- ◆ Import Data, with browse facilities to import from a flat file or an XML file.
- ◆ Import Data from a JDBC Data Source.
- ◆ Define Mapping (for flat files and JDBC data sources only), with a preview of the property-column mapping and editing facilities.
- ◆ Edit Data Model.
- ◆ Select Diagram Type, with a choice of ways to arrange the diagram.
- ◆ Select Layout Theme, with a choice of customized layouts.

This page only appears if you have chosen the Layout Diagram type.

See also *Graph Layout wizard*.

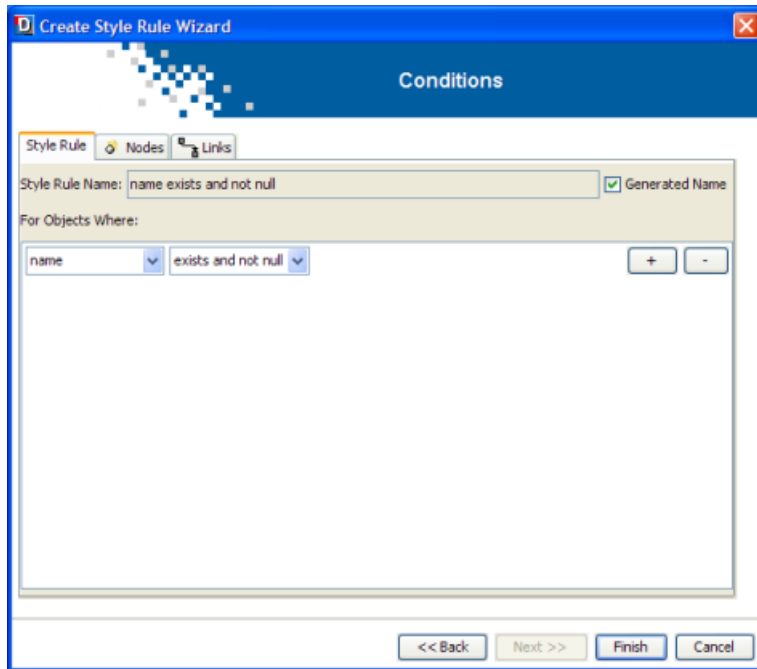
---

## Create Style Rule wizard

The Create Style Rule wizard simplifies the definition of data rules which can include attribute conditions.

To open the Create Style Rule wizard, choose **Edit>Create Style Rule** or right-click a rule and choose Create Style Rule in the shortcut menu.

Whenever you create (or change) a style rule, a Welcome page displays. If you do not want to see it pop up each time, uncheck the box Show Welcome page at the bottom left of the page. Following the Welcome page, the Conditions page displays, where you can define the conditions of the rule. See the following figure:



*The Conditions page*

---

## Node and link rules

In the Conditions page, the style rule name will be automatically computed from the conditions that you define in the pane below. You can however customize this name by unchecking the Generated Name option and typing your preferred name in the text field. Another option is to rename a rule in the Style Rules tree: right-click the rule and select Rename in the shortcut menu. Please note that you can at any time switch back to the generated name by checking the Generated Name box.

---

## Conditions

You can define conditions for user-defined types, states, and attributes:

- ◆ **User-defined type:** the object subtype to which the rule will apply, according to the types defined in your data model.
- ◆ **State:** the runtime state to which the rule will apply: selected, expanded, collapsed, in the legend, or level of detail (high, medium, or low). Expanded, and collapsed can only apply to nodes.
- ◆ **Attribute:** the name of an attribute of the object in your data model, for which a specific value is tested.

You can set a more or less restrictive rule by adding more or less conditions. The plus sign (+) allows you to add a condition. The minus sign (-) removes the corresponding condition. The more conditions a rule has, the more it is specific and the higher its priority.

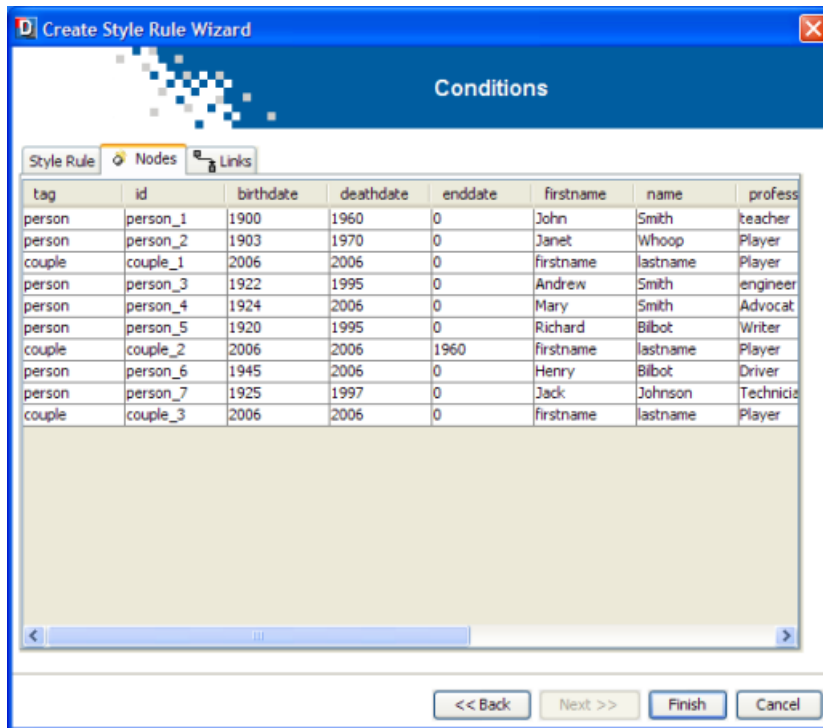
An attribute condition tests one or more attributes against a value. If you select an attribute, you must then set a condition on its value by selecting an operator from the second drop-down list and entering a value. You can set several attribute conditions.

The operator drop-down list reflects the type of the attribute: integer, real or string. For integer or real attributes, the operator menu offers *equals*, *does not equal*, *greater than*, *greater than or equal to*, *less than*, *less than or equal to*. For string attributes, the operator menu offers *equals*, *does not equal*, *contains*. You can also simply test the presence of an attribute in the data model (*exists* and *not null*).

---

## Viewing the data model in the style rule wizards

The Create Style Rule Wizard and the Change Style Rule Wizard both display a table view of the data model in the Nodes and Links tabs of the Conditions page. The tables show the property names and the current value for each property, on separate tabs for nodes and links.



*A table view of the data model in a style rule wizard*

When you create or change rules in one of these wizards, you can view the data properties and values that you can use to set conditions by selecting the Nodes or Links tabs.

---

## Change Style Rule wizard

The Change Style Rule Wizard simplifies the changing of the conditions in data rules. The content is the same as in the Create Style Rule Wizard, see *Create Style Rule wizard*.

To open the Change Style Rule Wizard, choose **Edit>Change Style Rule** or right-click a rule and choose Change Style Rule in the shortcut menu.



---

## Graph Layout wizard

The Graph Layout Wizard allows you to select a customized graph layout that is appropriate to the purpose of the diagram. This wizard sets a layout and customizes some property settings.

To open the Graph Layout Wizard, click the wizard icon in the Styling Customizer for the Graph Layout option. This icon is the top button of the vertical toolbar to the left.

The same functionality is available as a page within the New Diagram Wizard, see *New Diagram wizard*.

The layout themes available are as follows:

- ◆ Dependency Diagram
- ◆ Direct Links
- ◆ Orthogonal Long Links
- ◆ Orthogonal Short Links
- ◆ Grid Diagram - Placement in Columns
- ◆ Grid Diagram - Placement in Matrix
- ◆ Grid Diagram - Placement in Rows
- ◆ Organization Chart Diagram
- ◆ Radial Tree Diagram
- ◆ Symmetric Diagram
- ◆ Tree Diagram
- ◆ Workflow Diagram

---

## Data wizard

The Data wizard allows you to modify the data on which a diagram is based. The styling rules set for the diagram are not affected by the changes.

To open the Data wizard, go to Diagram Editing mode and choose **Data>Edit Model**.

The Data wizard has the following pages:

- ◆ Welcome
- ◆ Edit Your Data Model

In this wizard you can:

- ◆ Add and remove rows, where each row represents an object in the model.
- ◆ Add, remove, and rename columns, where each column represents a property in the model.
- ◆ Modify property values.

When modifying the data model, be aware that model properties may already be used in the style rules defined for the diagram. If you change a property name, the affected style rules may become inconsistent. Therefore, when you try to remove or rename a property, a warning message will be displayed indicating the style rules that will be affected by the operation, if any. In case you need to rename a property, the preferred path is to create the new property and modify the style rules to depend on this new property. Once this is done, you can safely remove the previous property using the Data wizard.

---

## The Rule menu

This menu contains many rule management commands, as described in this section. To manage your style rules, right-click a rule and choose from the Rule Menu.

---

### Cutting

To cut a style rule, select the rule and right-click, then choose **Cut** on the Rule Menu.

If you have cut a rule, you can do a Paste or Paste Styling Properties action to use the information in a different rule.

You can undo cut actions, see *The Undo and Redo facilities*.

---

### Copying

To copy a style rule, select the rule and right-click, then choose **Copy** on the Rule Menu. Once you have copied a rule, you can do a Paste or Paste Styling Properties action.

---

### Pasting

Before pasting, you must copy a style rule. Once you have copied a style rule, select the target one and choose **Paste** on the Rule Menu. This action changes both the conditions and the styling properties of the target rule.

---

### Pasting styling properties

Before pasting, you must copy a style rule. Once you have copied a style rule, select the target one and choose **Paste Styling Properties** on the Rule Menu. This action leaves the conditions of the target rule unchanged and changes the styling properties.

---

### Deleting

To delete a style rule, select the rule and right-click, then choose **Delete** on the Rule Menu.

You can undo deletion, see *The Undo and Redo facilities*.

---

### Renaming

If you want to customize the appearance of the style rules, you can choose your own names for each rule.

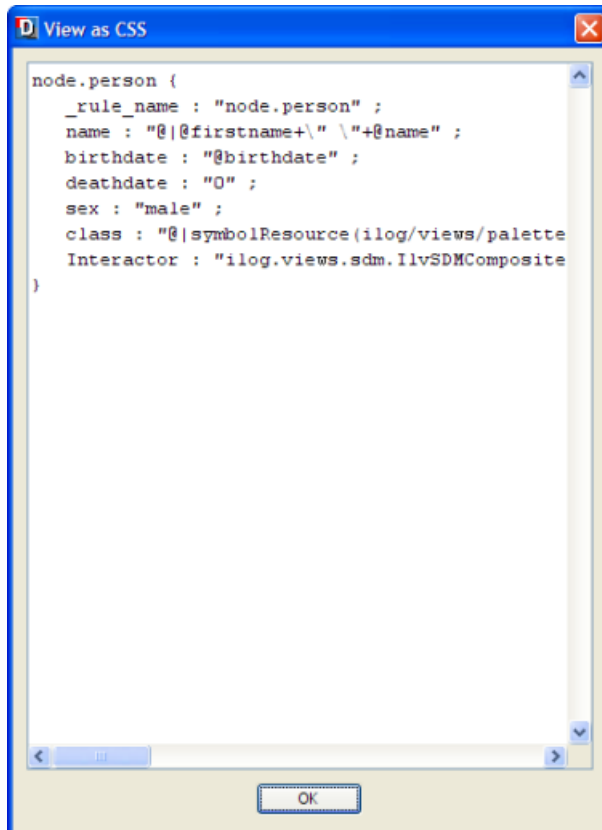
Select the rule and right-click, then choose **Rename** and enter your preferred name.

The name you enter appears in place of the dot-separated notation.

---

## Viewing as CSS

If you want to see the CSS code for a style rule, select the rule and right-click, then choose **View**.



```
node.person {
  _rule_name : "node.person" ;
  name : "@|@firstname+\\" \"+@name" ;
  birthdate : "@birthdate" ;
  deathdate : "0" ;
  sex : "male" ;
  class : "@|symbolResource(ilog/views/palette
  Interactor : "ilog.views.sdm.IlvSDMComposite
}
```

*An example of a rule in CSS*

Click OK to close the window.

The CSS code for all style rules and options is stored in the style sheet, which is saved as a .css file as part of the project.

---

## Moving up or down

You can reorder the style rules within a subtree. Order is important because a style rule that follows others may override property settings made by the preceding style rules. The order therefore represents a decision on priority.

**Note:** Overriding only occurs if the same property is set. If two rules set different properties, their order is unimportant.

To move a style rule up, select it and right-click, then choose **Move up** on the Rule Menu.

To move a style rule down, select it and right-click, then choose **Move down** on the Rule Menu.

Note that you cannot place a less specific rule below a more specific rule.

---

## Enabling or disabling

By default, all style rules defined are enabled.

To disable a style rule, select it and right-click, then choose **Disable** on the Rule Menu.

A disabled rule appears lighter and in italics. It is not applied (as if it had not been defined) but is available in the tree so that you can re-enable it later. Disabling and enabling rules is useful for testing purposes.

To re-enable a disabled style rule, select it and right-click, then choose **Enable** on the Rule Menu.

---

## The Undo and Redo facilities

The Undo command can apply to the following:

- ◆ In Style Editing mode:
  - Property settings
  - Rule changes (creation or modification using a wizard, enable/disable)
  - Cut or Delete operations done using the Rule Menu
  - Selecting/clearing an option in the Options dialog box displayed when you choose **Edit>Options**
- ◆ In Diagram Editing mode:
  - Object moves
  - Object creation
  - Object deletion
  - Property changes in the Data Properties view

To undo one of these actions, choose **Edit>Undo**.

There are unlimited levels of Undo.

When you have undone an operation, you can redo it by choosing **Edit>Redo** .

---

## The Check Rules facility

The Check Rules facility checks for:

- ◆ Bad property values (wrong type of value)
- ◆ References to nonexistent data properties
- ◆ Conditions that do not currently apply to any object in the data, indicating rules that may be useless (however, such rules may be useful with a different data set and so do not always constitute an error)

To check your rules, choose **Edit>Check Rules** or click the Check Rules button in the Messages panel.

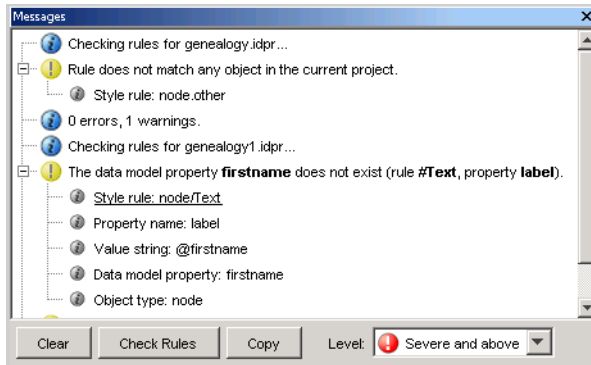
The result of the check appears as a series of messages in the Messages panel, see *Messages for troubleshooting*. Note that messages of all levels are displayed, regardless of the filter level set in the Messages panel.

---

## Messages for troubleshooting

The Designer provides a Messages panel which you can show or hide. This panel is dockable in any position within the main panel or can be a separate window.

The Messages panel displays a list of messages, see the following figure.



### *The Messages panel*

You can filter the list according to the severity level as shown. By default, only messages of level Severe and above are shown.

The messages help you to debug your style rules.

If you expand a message, you can see the details of the style rule and styling property value that caused the error. You can double-click the style rule listed below the message to highlight the following:

- ◆ The rule in the Style rules hierarchy
- ◆ The affected nodes in the Diagram (if any)
- ◆ The property setting in the Styling Customizer (if the cause was a property setting)



---

## Printing the diagram

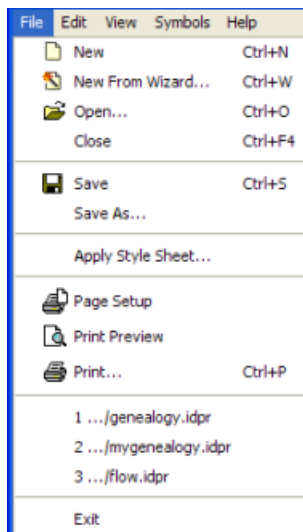
In the Designer, you have the following printing facilities for the diagram:

- ◆ Page setup
- ◆ Print preview
- ◆ Print (one, all or selected pages)

---

### Print operations

Print operations are available from the *File* menu, see the following figure.



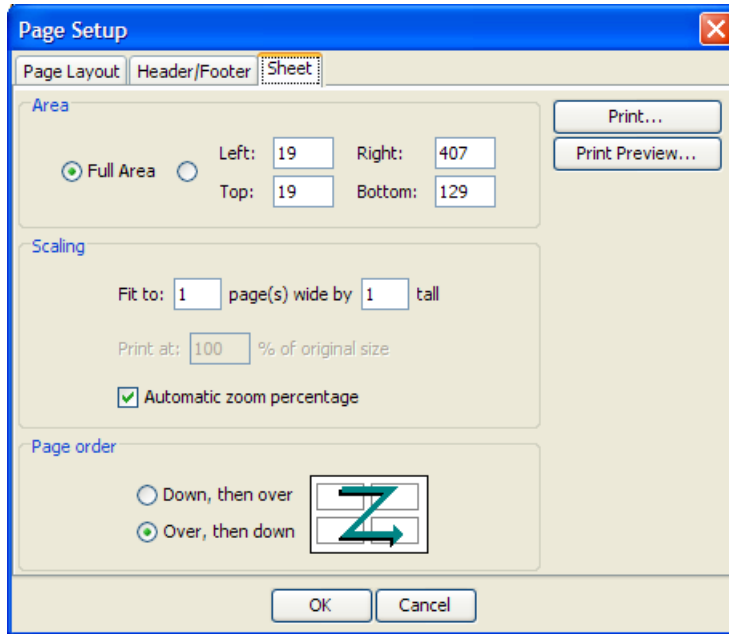
*The print options on the File menu of the Designer*

To print the diagram to the default printer from the application, click the Print button in the Print Preview window or choose **File>Print**. You can print to a printer or to file.

---

### Page Setup

The Page Setup window offers extensive setup facilities, see the following figure.



### *Page Setup facilities*

In the Page Layout tab, you can select Portrait or Landscape and set paper size and margins. The Designer automatically sets Portrait or Landscape initially according to the shape of the diagram.

The Headers/Footers tab gives you a choice of headers and footers and allows you to define a custom Header and Footer, setting a specific Font, if you want to.

In the Sheet tab, under Scaling, you can split a large diagram into separate pages for printing, for example, by specifying Fit to 3 page(s) wide by 2 tall, to get six pages.

In the Sheet tab, under Page order, you can specify the order in which the pages are numbered and printed:

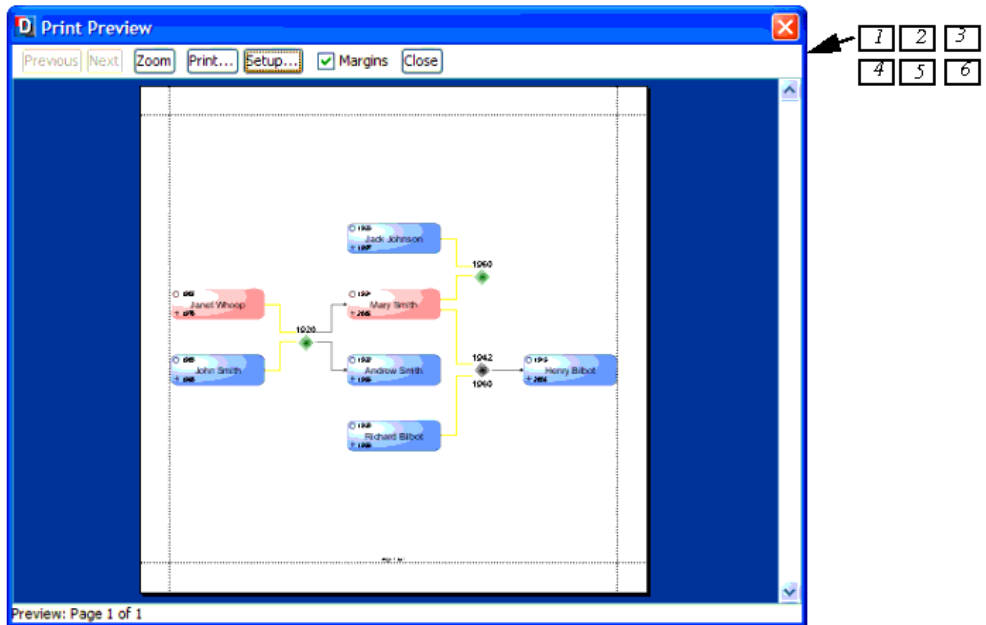
- ◆ *Down, then over*: prints top to bottom, so for six pages, it prints 1, 2 down then 3, 4 down then 5, 6 down
- ◆ *Over, then down*: prints left to right, so for six pages, it prints 1, 2, 3 across and then again 4, 5, 6 across

For each page, under Area, you can print either the full area or effectively zoom in to print part of the page by setting left, right, top, and bottom distances from the edge of the page. These distances are in pixels.

---

## **Print Preview**

The Print Preview window allows you to display each page specified in the page setup and view the pages in the order in which they will be numbered and printed.



*Print Preview: page 1 in a 3x2 setup with Over Then Down*

Note that you will only see the page numbers if you have specified a header or footer that contains them.

---

## Comparing the Designer and the API

The Designer provides a convenient visual environment for styling diagram components for use in a graphical application. The output is in the form of a project, comprising a project file and an associated style sheet (.css file). You are recommended to start styling in the Designer.

The examples supplied with the Designer make use of symbols. You can also open legacy diagrams containing general nodes or composite graphics in the Designer and style them.

General nodes are simpler objects: you may find them easier to customize. They are also faster to display, which can be of particular interest in the case of diagrams with a lot of graphics.

In the Java™ API, you can apply style sheets in the CSS for Java format to symbols, composite graphics, general nodes, and general links. You can load a project file to display a diagram.

In this release, not all the refinements available in the API are active in the Designer. The main limitations are as follows:

- ◆ Interactors: only the predefined ones are available
- ◆ Renderers: only the predefined ones are available
- ◆ Style rules: Child-of-parent object conditions are not supported

The decision points at which you may decide to use the Java API as well as the Designer concern the following:

- ◆ Custom data model
- ◆ Custom interaction
- ◆ Custom rendering
- ◆ Advanced graphic objects
- ◆ Custom printing

If your data cannot be connected by JDBC or is not in a flat file or XML, you must use the Java API to define a custom data model before starting development of your diagram component.

If, after you complete a development cycle in the Designer, you find that you require interaction or rendering that is not available in the Designer, you can achieve these by using the Java API directly.

If you need advanced graphic objects that are not available in the Designer, you can write your own graphic objects with the Java API of the JViews Framework. However, you will not be able to customize them with the Styling Customizer in the Designer.

If you need specialized printing facilities beyond the page and area facilities available in the diagram component, you can code these in the Java API.

# ***Using more Designer features***

Presents and gives examples of more advanced features of the JViews Diagrammer Designer (those not presented in *Getting started*).

## **In this section**

### **Changing Layout Parameters**

Explains how to use the Styling Customizer for graph layout.

### **Using the Graph Layout Wizard**

Explains how to select a graph layout in the wizard.

### **Loading a different style sheet**

Explains how to change the style sheet associated with the diagram.

### **Resetting a default value after a state change**

Explains how to handle values set by default that change because of a temporary state.

### **Importing data from a flat file**

Explains how to import data from a text file and map it.

### **Inserting an icon**

Explains how to load an image file and insert it into a node.

### **Creating a user-defined data model**

Explains how to create a simple network model based on the in-memory data in the New Diagram Wizard.

### **Adding decorations for alarm severities**

Explains how to configure nodes so that they dynamically reflect any change in their status.

**Adding a node to a diagram**

Explains how to insert a node on an existing link and then lay out the nodes again.

**Using subgraphs**

Explains how to create a group, which is a node that contains a subgraph of other nodes.

**Displaying different levels of detail**

Shows how to define three different levels of detail for displaying the nodes.

**Using background maps**

Describes how to load a background map into a diagram.

**Importing data from a database**

Explains how to load data from a database and map it.

**Using data model expressions**

Explains an advanced feature for inserting references to the data model or expressions in certain property values.

# ***Changing Layout Parameters***

Explains how to use the Styling Customizer for graph layout.

## **In this section**

### **Customizing graph layout algorithms**

Gives a short description of how to customize a layout algorithm.

### **Using the genealogy diagram**

Explains how to access the genealogy example used in this exercise to perform graph layout customizations.

### **Changing a basic parameter**

Explains how to change the flow direction of the diagram.

### **Changing a link style parameter**

Explains how to change the shape of the links in the diagram.

### **Setting a link style with mixed links**

Explains how to specify a different shape for some specific links in the diagram.

---

## Customizing graph layout algorithms

Graph Layout algorithms can be customized by setting various parameters. These appear in the Styling Customizer for the Graph Layout option. This customizer also gives you access to a wizard for selecting an appropriate layout theme. A layout theme is effectively a set of parameter settings applied to a specific graph layout.

Graph Layout algorithms have many parameters, but you do not usually need to vary many of them to obtain the required results.



---

## Using the genealogy diagram

You will be working with the genealogy example which you created in *Getting started*. The diagram for this example is laid out according to a Graph Layout algorithm.

### To open the genealogy example if it is not already open:

1. Choose **File>Open**.
2. Select the appropriate project file.

There is also a version supplied as `<installdir>/jviews-diagrammer86/bin/designer/data/examples/genealogy.idpr`.

**Note:** You can only use **File>Open** to open project files, not any other type of file.

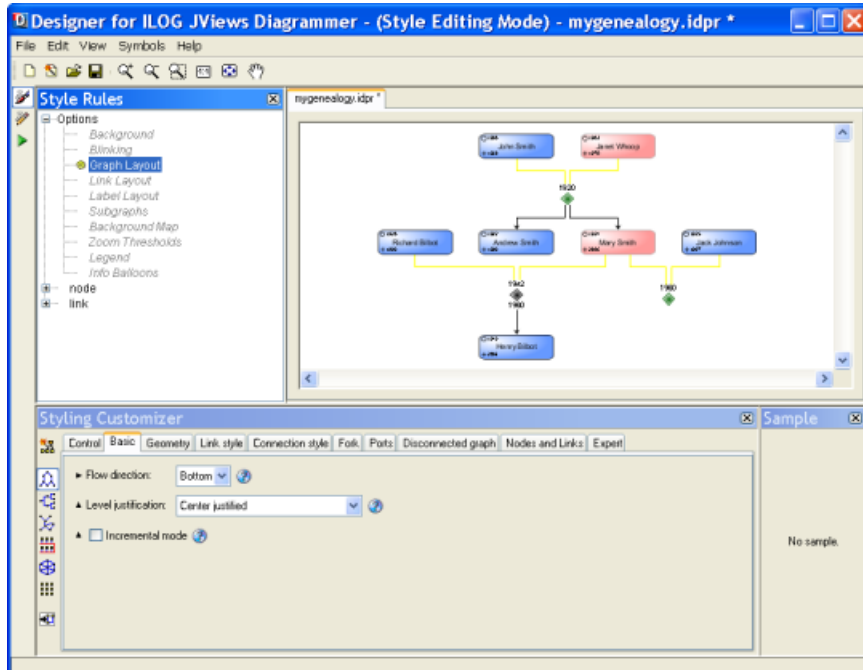
## Changing a basic parameter

The Basic tab in the Styling Customizer contains the parameters of the selected graph layout that you are most likely to want to change.

### To change the orientation of the diagram:

- ◆ In the Basic tab, set Flow Direction to `Bottom`.

See the orientation of the diagram change.



*Specifying flow direction*

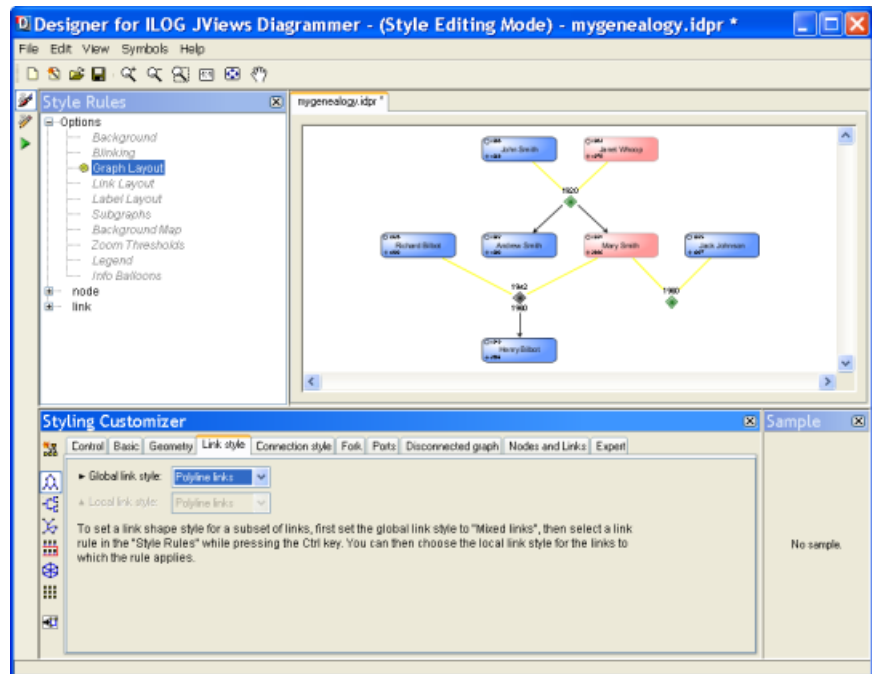
## Changing a link style parameter

The Link Style tab contains the parameters that govern the geometric form of the links.

### To change the form of the links:

1. In the Basic tab, set the Global link style to Polyline links.

See the form of the links change.



*Specifying polyline links*

2. Compare figure *Specifying polyline links* with the orthogonal links in figure *Specifying flow direction*.

---

## Setting a link style with mixed links

The Styling Customizer for the graph layout contains various layout parameters that apply to individual nodes and links or subsets of nodes and links.

In order to specify such parameters, you first need to create a rule to specify the nodes or links, then you select Graph Layout under Options and select the new node or link rule while holding the CTRL key.

### **For example, to change the Link Style parameter:**

1. In the Style Rules, select Graph Layout under Options.
2. In the Link style tab, set `Global link style` to `Mixed links`.

This setting allows you to mix link styles in the diagram; otherwise the specified global link style is applied to all links.

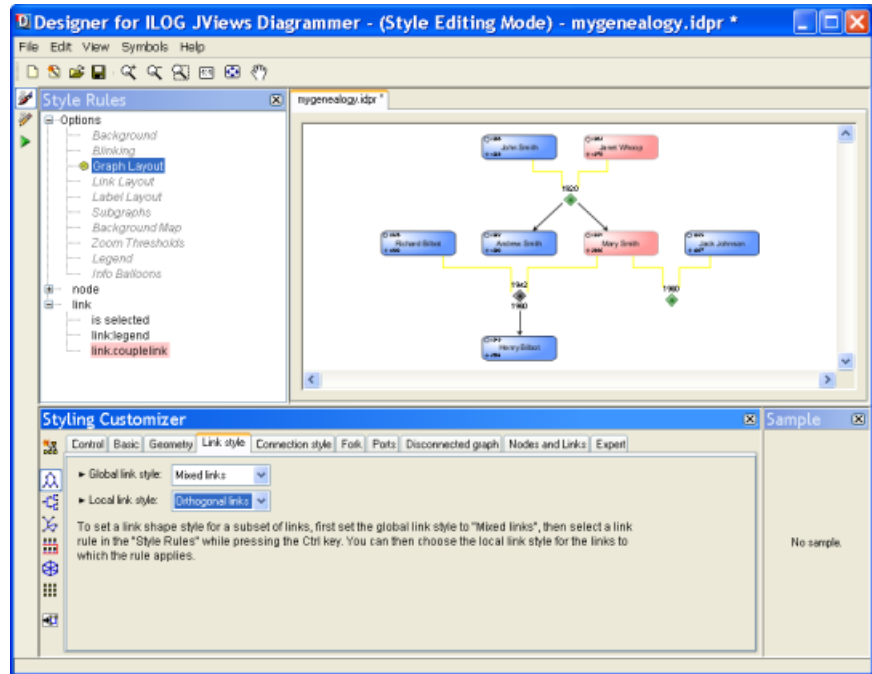
3. Press the CTRL key and, while holding it, select the `link.couplelink` rule.

At this point, two rules should be selected:

- ◆ The Graph Layout option, indicated by a blue background
- ◆ The `link.couplelink` data rule, indicated by a pinky-red background

4. In the Link style tab, you see that `Local link style` is enabled. Change it to `Orthogonal links`.

See the form of some links change.



*Mixed links with different link styles*

**To close the diagram:**

1. Choose **File>Close**.

You are prompted to save your changes.

2. Click No so that you do not overwrite the previous settings.

---

## Using the Graph Layout Wizard

You can use the Graph Layout Wizard to set up a complete look and feel for the layout of a diagram instead of setting parameters singly.


Continue with the genealogy example which you customized in *Changing Layout Parameters* or choose **File>Open** and select `<installdir>/jviews-diagrammer86/bin/designer/data/examples/genealogy.idpr` if it is not already open.

**Note:** You can only use **File>Open** to open project files, not any other type of file.

The diagram is laid out according to a hierarchical Graph Layout algorithm, with several custom parameter settings.

### To set up a complete look and feel for the layout of the diagram:

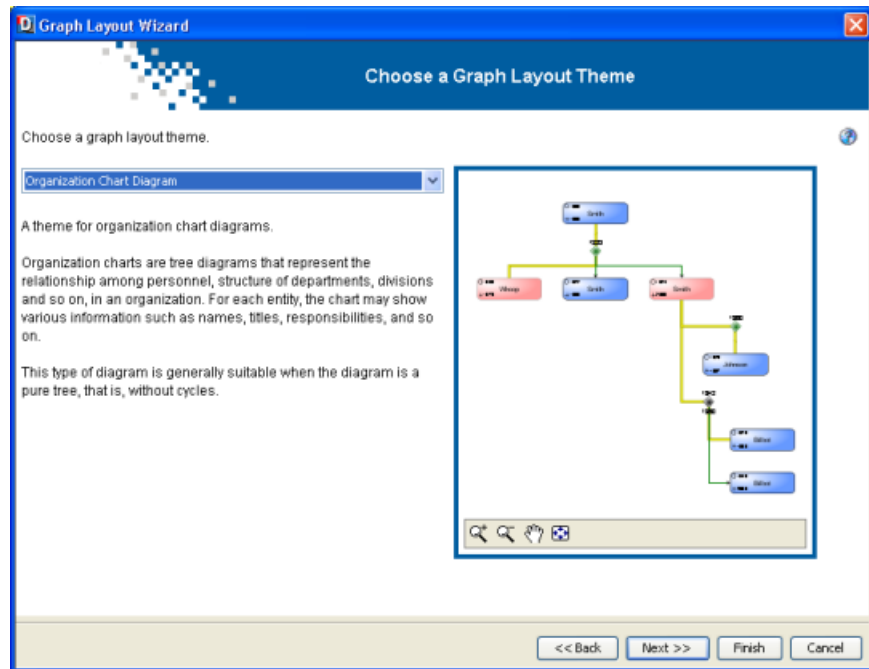
1. In the Styling Customizer for the graph layout, click the top button in the vertical

toolbar to the left .

This opens the Graph Layout Wizard.

2. Click Next.
3. Choose **Organization Chart Diagram**.

You can see a preview of the diagram to the right, see the following figure.



### *Choosing an organization chart layout*

4. Click Finish.

You do not need to know the details of the parameter settings if you set up the layout in this way. But in case you are interested, you can look at the Styling Customizer with Graph Layout selected to see the property settings, for example, the layout type for the Organization Chart is Tree.

5. Choose **File>Close** to close the diagram.

You are prompted to save your changes.

6. Click No so that you do not overwrite the previous settings.

---

## Loading a different style sheet

The style sheet saved as part of a project can be changed without changing the data. Note, however, that if styling is applied according to attributes of the data model, a different style sheet may not be appropriate for the data.

Open the genealogy example which you created in *Getting started* by choosing **File>Open** and selecting the appropriate project file if it is not already open.

**Note:** You can only use **File>Open** to open project files, not any other type of file.

### To load a different style sheet:

1. Choose **File>Apply Style Sheet** and browse to select one of the `genealogy` style sheets.

These effectively represent different stages of customization and all can be applied to the genealogy data. If you select a style sheet that was not designed for the data, the effects are unpredictable.

A dialog box displays asking you if you want to replace the existing style rules with the new style sheet or if you want to append the new style sheet to the existing rules.

2. Click Replace.
3. Choose **File>Close** to close the diagram.  
You are prompted to save your changes.
4. Click No so that you do not overwrite the previous settings.



---

## Resetting a default value after a state change

You may have noticed that the arrow heads on the couple links in the genealogy example are not the same color as the rest of the link. The arrowhead color is a separate styling property of links, `Color` in the Arrow tab.

When a user selects a link, the rule that determines the color is the `link:is selected` rule. This rule makes the link color orange, but does not affect the arrowhead.

Open the genealogy example which you created in *Getting started* by choosing **File>Open** and selecting the appropriate project file if it is not already open.

**Note:** **File>Open** can only open project files.

### To include the arrowhead in the color change:

1. Select the `link:is selected` rule.
2. In the Arrow tab, set `Color` to orange.
3. Switch to Preview Mode to test the result.
4. Select a couple link and see the color change to orange, arrowhead included.
5. Select a node instead and see the link color revert to green.

But the arrowhead does not revert to green!

This result is at first confusing but has a simple explanation and solution.

The `Color` property for the arrowhead is set to a default value (black) when the link is created. Therefore black is the color you see initially for the arrowheads on couple links. If the user (you, in this example), selects a link and then deselects it, there is no rule explicitly giving the arrowhead a new color, and so it stays orange.

### To make sure the arrowhead is set back to green:

1. Select the generic `link` rule.
2. In the Arrow tab, set `Color` to green.

It is only necessary to (re)set the default value explicitly when the styling property undergoes a temporary change due to the state of an object. The possible states are `selected`, `expanded`, `collapsed`, or `legend` for nodes and `selected` or `legend` for links.

Note that not all properties have default values.



# *Importing data from a flat file*

Explains how to import data from a text file and map it.

## **In this section**

### **Loading the file**

Explains how to load the flat file from the New Diagram wizard.

### **Importing node data**

Explains how to import the node data of the flat file.

### **Importing link data**

Explains how to import the link data of the flat file.

### **Mapping node data**

Explains how to map properties to node data.

### **Mapping link data**

Explains how to map properties to link data.

### **Editing the data model**

Explains how to edit the flat file data model.

### **Selecting the diagram type**

Explains how to select the diagram type.

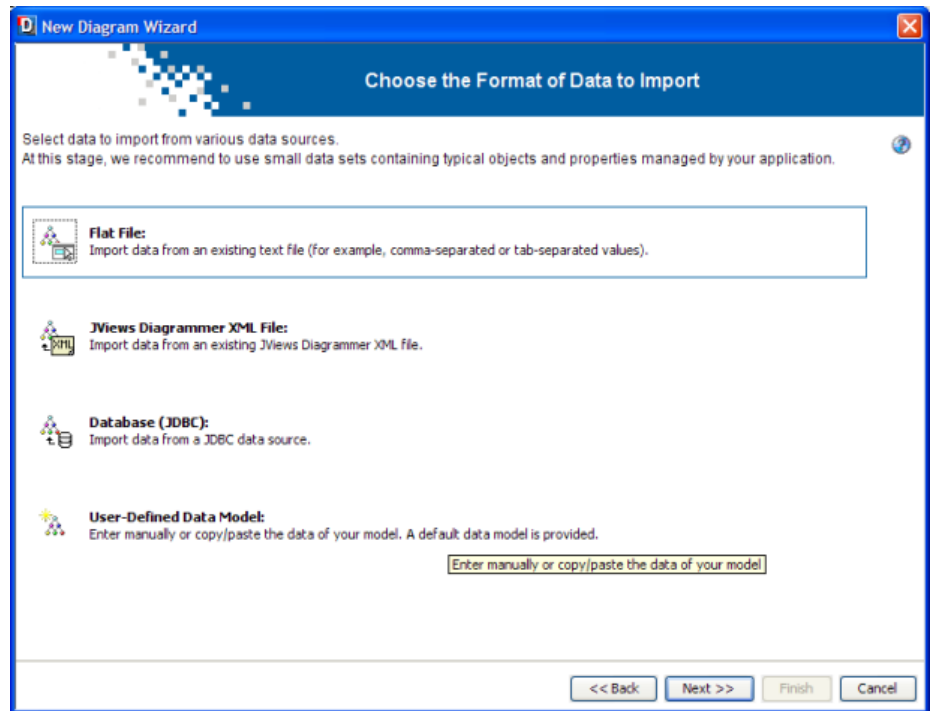
## Loading the file

The flat file data source allows you to import data from any flat file including a CSV (comma-separated values) file.

### To import a flat file:

1. Choose **File>New from Wizard**.
2. Click Next.
3. Select the Flat File option.

See the following figure.



*Importing data from a flat file*

4. Click Next.

This section makes use of the Arizona example.

You will need to import data for nodes and links separately.

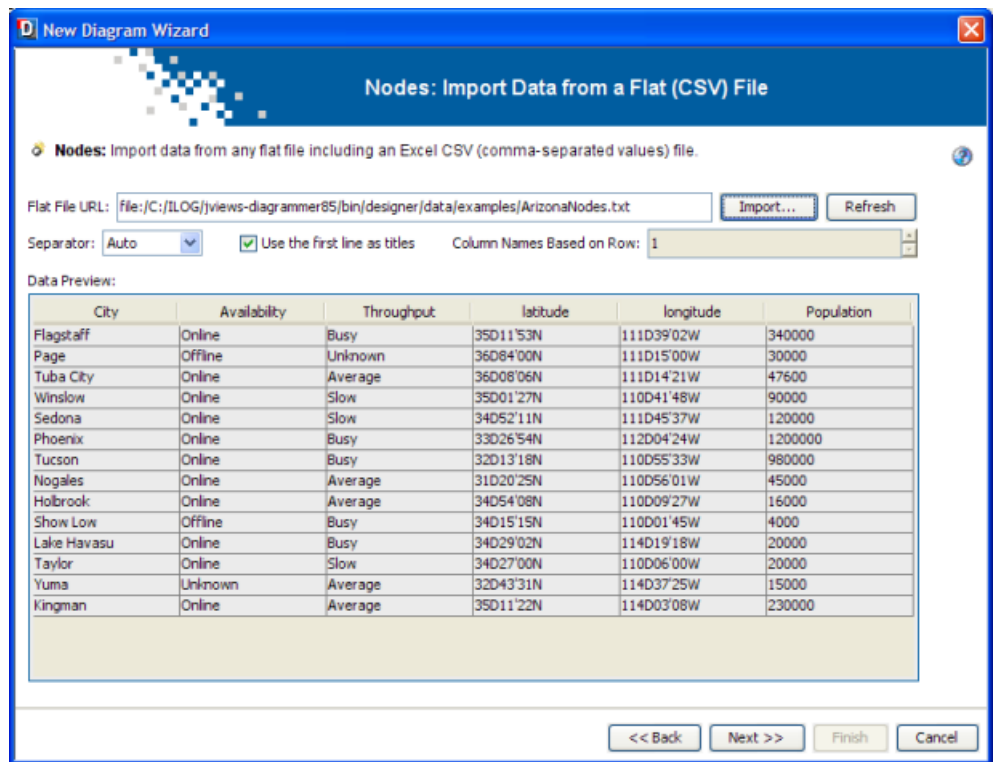
## Importing node data

You can choose a separator character and the number of the row containing the column names. The separator character is the character that appears between data values on rows. If you do not change the default settings: Auto and 0 (zero) respectively, the Designer tries to interpret the file automatically. If it finds a known separator character, it updates the File Preview on the left.

### To import the node data:

1. Browse to the `ArizonaNodes.txt` file.
2. Click Open.

See the following figure.



*Importing data on places in Arizona as nodes*

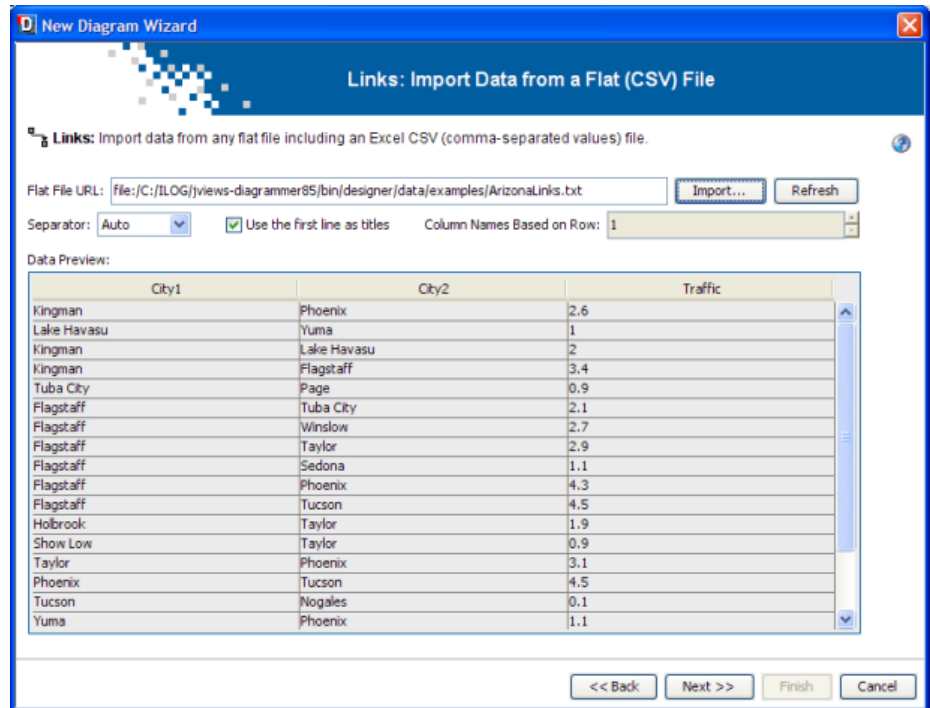
## Importing link data

You can choose a separator character and the number of the row containing the column names. The separator character is the character that appears between data values on rows. If you do not change the default settings: Auto and 0 (zero) respectively, the Designer tries to interpret the file automatically. If it finds a known separator character, it updates the File Preview on the left.

### To import link data:

1. Switch to the Links pane.
2. Browse to the `ArizonaLinks.txt` file.
3. Click Open.

See the following figure.





*Importing data on traffic between places in Arizona as links*

4. Once you have imported data for nodes and links, click Next.

---

## Mapping node data

Properties that are not mapped to node data and require to be mapped appear with the following status  in the list of properties, and the same status appears on the Nodes tab. This is the case of the `id` property. You must map it to a column because its values should uniquely identify the rows.

Additional properties that are not mapped to node data appear with status  in the list of properties.

### To map the `id` property to the `City` column:

1. Click the `id` property.
2. Click the `City` column.


You can map other properties as required.

In this example, you have georeferenced data, so you need to use the `latitude` and `longitude` columns to position the nodes. These columns are mapped automatically because of their names. The preferred format for latitude and longitude is DMS (degrees, minutes, seconds). For example: 32D43'31"N. If the `latitude` and `longitude` columns contain only numbers, the values are interpreted as degrees. For example: `latitude` 12.56 is interpreted as 12.56 degrees, which equals 12D33'36"N (in DMS).

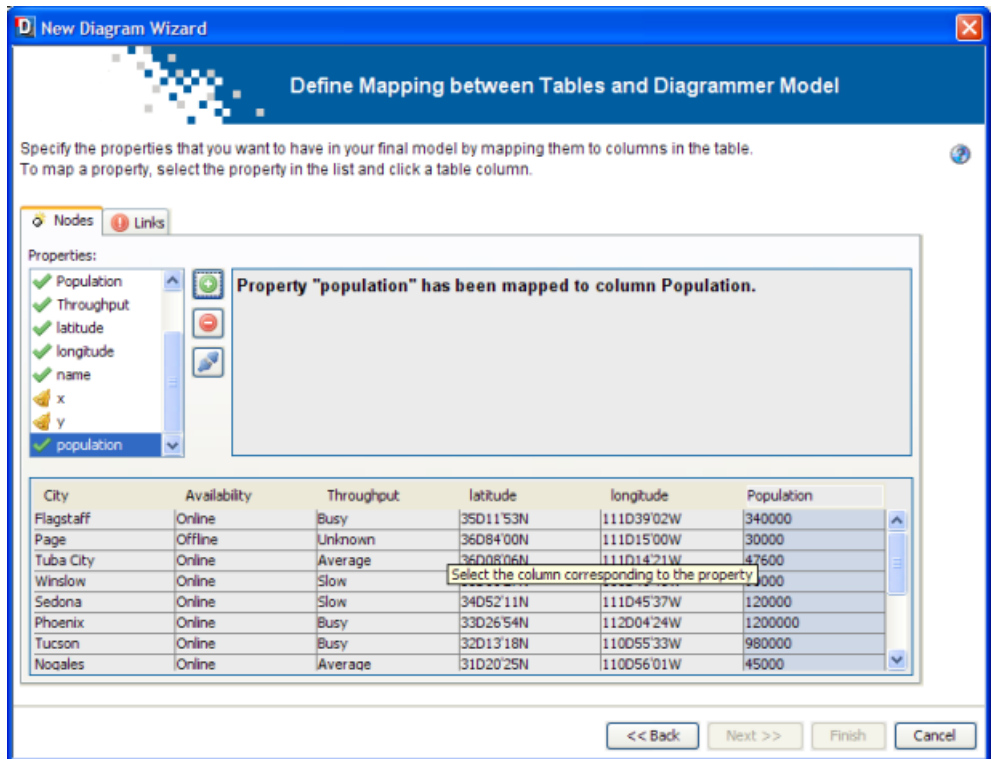
### To map several properties to the same column:

- ◆ Map both the `id` property and the `name` property to the `City` column.

### To add properties and map them to columns:


1. Click the  button.
2. Type the property name `population`.
3. Map this new property to the `Population` column.

The result should be as in the following figure. Note that with `City` mapped to `name`, the name appears on the nodes in the preview.



*Mapping node data to properties*

**To remove the mapping of a property:**

1. Select the property in the list.
2. Click the  button.



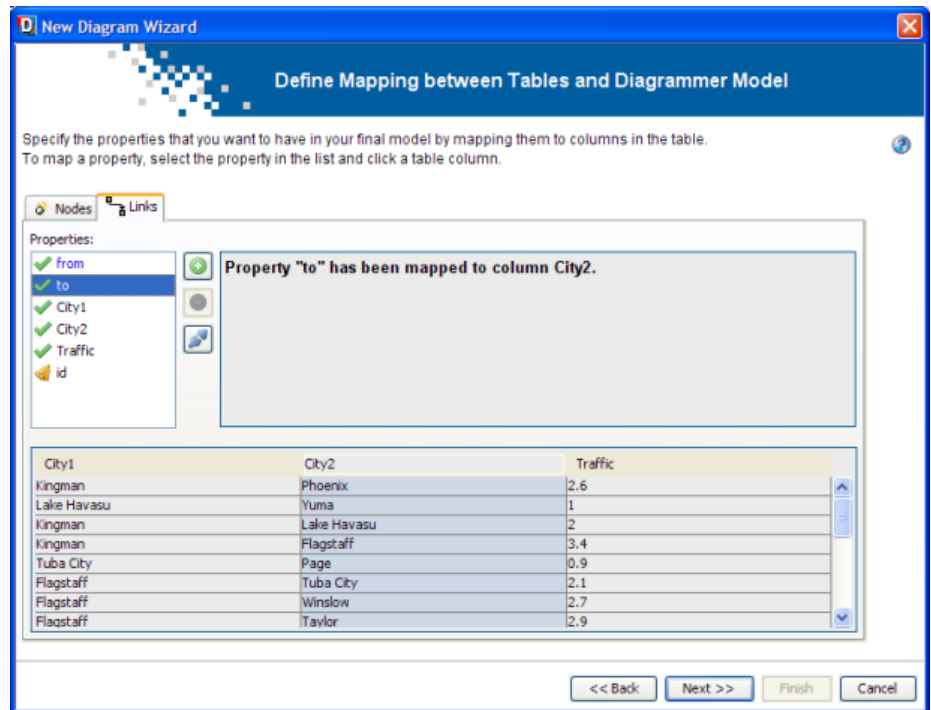
## Mapping link data

You must map the `to` and `from` properties to columns because these indicate the start and end points of the links. You can map other properties as required.

### To map the `to` and `from` properties:

1. Click the `from` property then click the column `City1`.
2. Click the `to` property then click the column `City2`.

In this example, the `id` property is not mapped.



*Mapping link data to properties*

3. Once you have completed the mapping for both nodes and links, click Next.

---

## Editing the data model

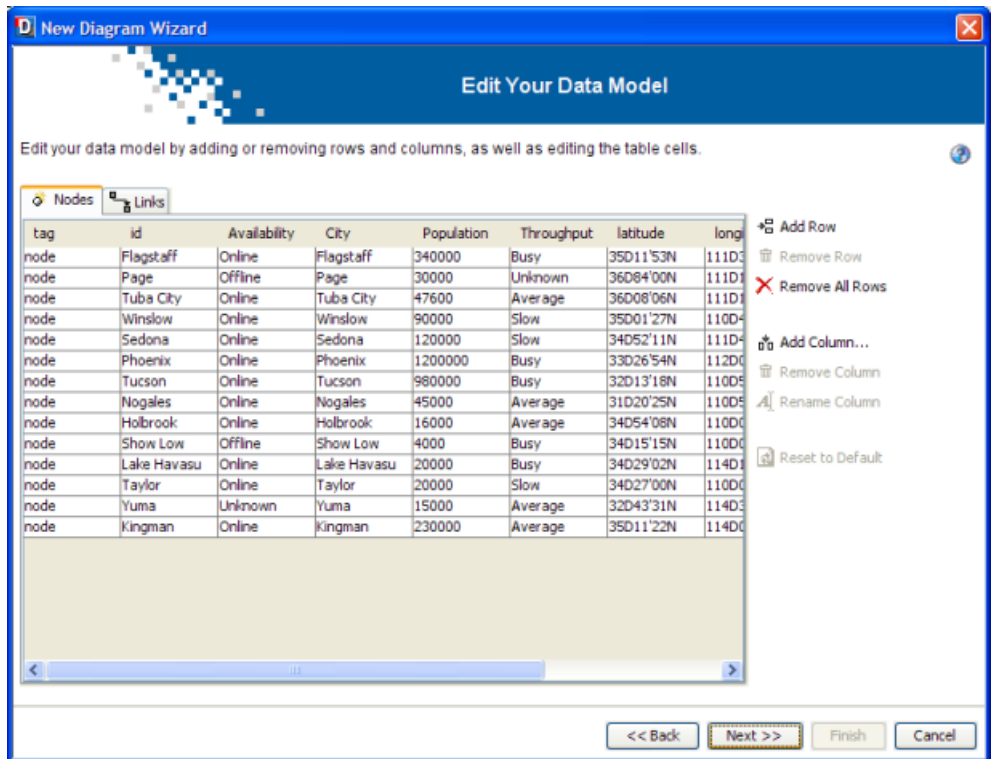
You can edit the basic data model to:

- ◆ Add and remove rows (a row represents an object in the data model)
- ◆ Add, remove, rename columns (a column represents a property of the objects in the data model)
- ◆ Change cell values (a cell represents a property of a specific object)
- ◆ Remove all objects
- ◆ Restore the data model to its initial state
- ◆ Paste data from the clipboard

If you click **Remove All Rows**, you clear all the data values. To retrieve the values you saw initially, click **the Reset to Default** button.

Properties `id` and `tag` are mandatory for the SDM model and therefore cannot be removed or renamed by the user.

**Paste from Clipboard** replaces the contents of the current table by the information present in the clipboard. It asks if the user wants the column names to be named based on the first row or not. If so, the first line of data is used to give the column names. If not, the column names are named as A, B, C, and so on, and can later on be renamed by the user manually.



### *Editing the flat file data model*

You will keep this data model as it is.

#### **To validate your data model:**

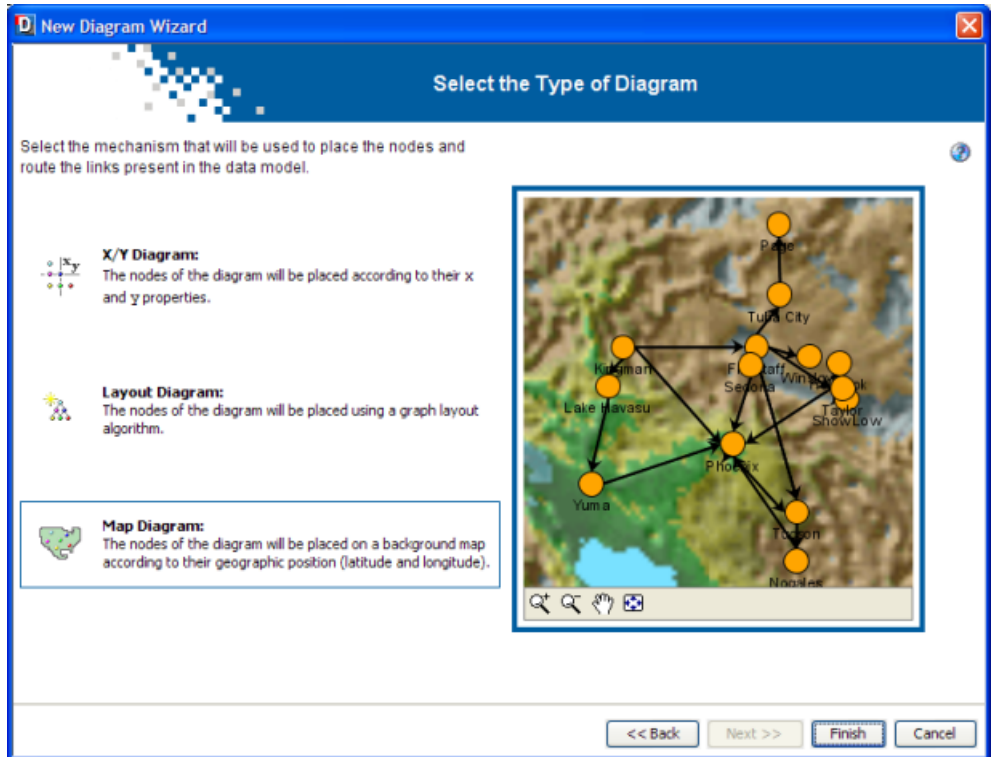
- ◆ Click Next.

## Selecting the diagram type

You need to select a diagram type for the correct positioning of the nodes.

As you have latitude and longitude mapped, the Map Diagram is selected by default, see the following figure.

**Note:** If the JViews Maps product is not installed, the X/Y Diagram is selected by default. Click Finish.



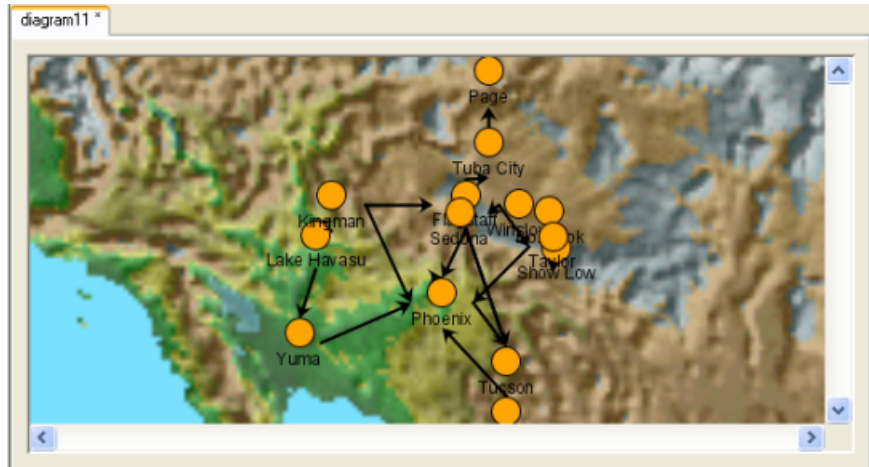
*Specifying a map diagram*

**To select the Map diagram:**

1. Click Finish.

If you need to make changes, you can click Back instead to return to specific wizard pages.

The resulting diagram is as shown in the following figure.



*The initial Arizona physical diagram*

2. Keep the diagram open to proceed to the next section or save it by choosing **File>Save As**.



# *Inserting an icon*

Explains how to load an image file and insert it into a node.

## **In this section**

### **Image file formats**

Lists the image file formats supported by the Designer.

### **Setting a special symbol**

Explains how to apply a special symbol to the nodes of the diagram.

### **Creating rules for population ranges**

Explains how to create style rules associated with different images to represent population ranges.

### **Rule priority**

Explains how to interpret the style rules tree in terms of priority.

---

## Image file formats

You can customize a special symbol to contain an image file.

If you have icons that you want to use to represent your objects, you load the image files through the Styling Customizer.

The image file formats supported by the Designer are as follows:

- ◆ GIF
- ◆ JPG
- ◆ PNG

In the Arizona example, three images are used according to city population:

- ◆ For cities with fewer than 100 000 residents: `little.gif`
- ◆ For cities with more than 1 000 000 residents: `biga.gif`
- ◆ For the remaining cities (between 100 000 and 1 000 000 residents): `lightBulb.gif` )




---

## Setting a special symbol

To be able to load image files to represent your nodes in the Arizona physical diagram, you need first to assign a special symbol to the default graphic representation of the nodes.

**To set a special symbol to the nodes:**

1. Click the node rule in the Style Rules tree.
2. Click the Assign symbol button at the bottom of the Styling Customizer.
3. Select the IBM® ILOG® icon  in the Basic category of the Shared Symbols palette.
4. Click Apply.

---

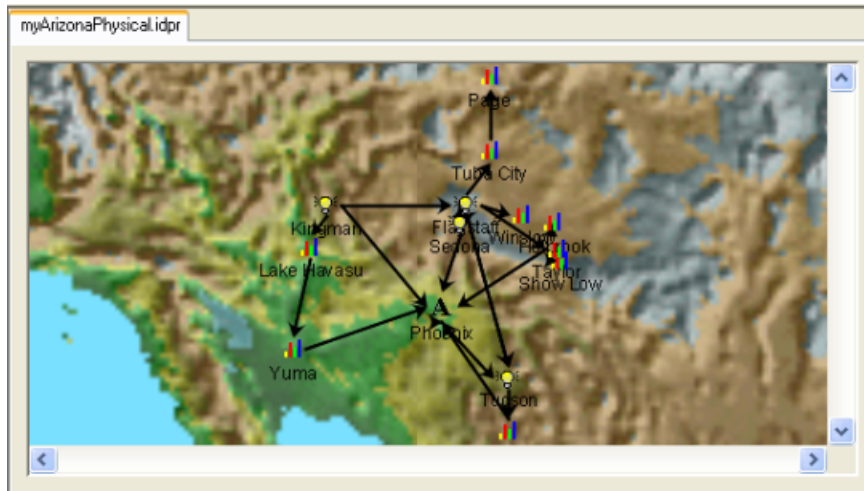
## Creating rules for population ranges

You are going to identify the cities with a different graphic representation according to three specific population ranges.

You need to define three style rules to specify the three conditions on population:

1. Select the node rule and choose **Edit>Create Style Rule**.
  - a. In the Conditions page, define the condition `population exists and not null`.
  - b. Click Finish.
  - c. With the new `population exists and not null` rule selected, in the Styling Customizer, type the following path in the `image` parameter field: `file:./data/examples/lightBulb.gif`.
  - d. Press ENTER.
2. With the `population exists and not null` rule selected, choose **Edit>Create Style Rule**.
  - a. In the Conditions page, click the + sign and define another condition: `population is less than 100000`.
  - b. Click Finish.
  - c. Rename the rule to `population<100000`.
  - d. With the new `population<100000` rule selected, in the Styling Customizer, type the following path in the `image` parameter field: `file:./data/examples/Little.gif`.
  - e. Press ENTER.
3. Select the `population exists and not null` rule and create the third style rule:
  - a. Choose **Edit>Create Style Rule**.
  - b. Click the + sign and define the condition `population is greater than 1000000`.
  - c. Click Finish.
  - d. Rename the style rule to `population>1000000`.
  - e. With the new `population>1000000` rule selected, type the following path in the `image` parameter field of the Styling Customizer: `file:./data/examples/big.gif`.
  - f. Press ENTER.

The result should be as shown in the following figure.



*The use of images in the Arizona example*

---

## Rule priority

Note that the style rules are displayed in a tree which represents a progressive refinement of styling. As well as the meaning within each style rule, there is also meaning in the tree structure. In fact, the style rules are displayed in ascending order of specificity (least specific at the top of the tree).

The `population<100000` rule is more specific than the `population exists and not null` rule. Therefore the `population<100000` rule has priority. The settings in the `population<100000` rule override settings in the node rule for all nodes that have population less than 100000.

You can move rules up and down the tree to change their priority, but you cannot place a more specific rule above a less specific one. The commands to move rules are on the Rule Menu which is available when you right-click a rule.

# ***Creating a user-defined data model***

Explains how to create a simple network model based on the in-memory data in the New Diagram Wizard.

## **In this section**

### **Creating the data model**

Explains how to create a user-defined data model from the New Diagram wizard.

### **Editing the data model**

Explains how to edit the default user-defined data model to create a network model.

### **Changing the graphic representation of the nodes**

Explains how to customize the default representation of nodes.

---

## Creating the data model

The user-defined data model allows you to start from in-memory data in Java™ classes compliant with the data model requirements of JViews Diagrammer.

**To create a data model in memory interactively:**

1. Choose **File>New from Wizard**.
2. Select the user-defined data model option (default option).

You start from a basic data model containing four nodes and four links, see *Editing the in-memory data model*.

Usually, data models contain either x and y data, or latitude and longitude data. Here the data model contains both.

---

## Editing the data model

You can edit the basic data model to:

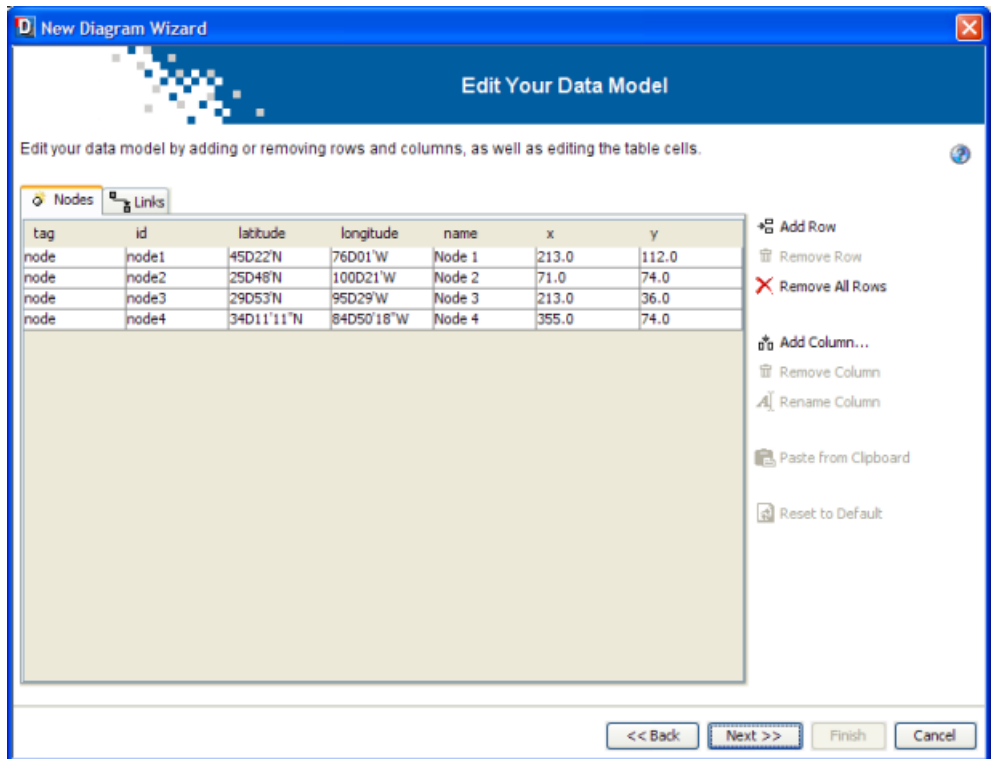
- ◆ Add and remove rows (a row represents an object in the data model)
- ◆ Add, remove, rename columns (a column represents a property of the objects in the data model)
- ◆ Change cell values (a cell represents a property of a specific object)
- ◆ Remove all objects
- ◆ Restore the data model to its initial state
- ◆ Paste data from the clipboard

For example, you could remove the x and y columns if you do not want an X/Y Diagram. Note that you can create a Layout Diagram without any specific layout attributes.

If you click **Remove All Rows**, you clear all the data values. To retrieve the values you saw initially, click **the Reset to Default** button.

Properties *id*, *tag*, *from* and *to* are mandatory for the SDM model and therefore cannot be removed or renamed by the user.

**Paste from Clipboard** replaces the contents of the current table by the information present in the clipboard. It asks if the user wants the column names to be named based on the first row or not. If so, the first line of data is used to give the column names. If not, the column names are named as A, B, C, and so on, and can later on be renamed by the user manually.

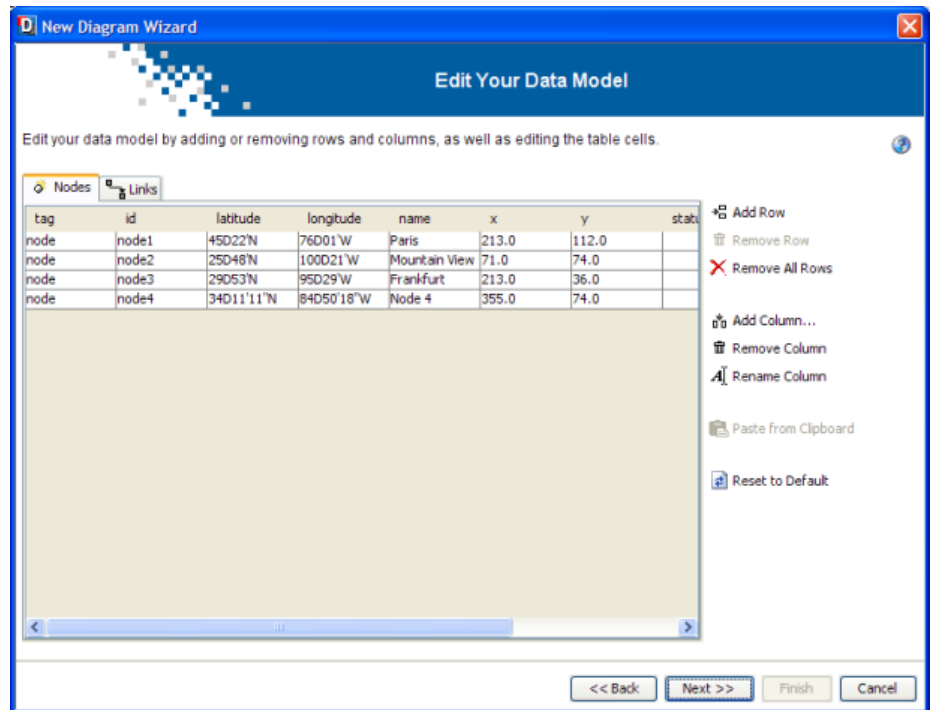


*Editing the in-memory data model*

**To create a simple network model:**

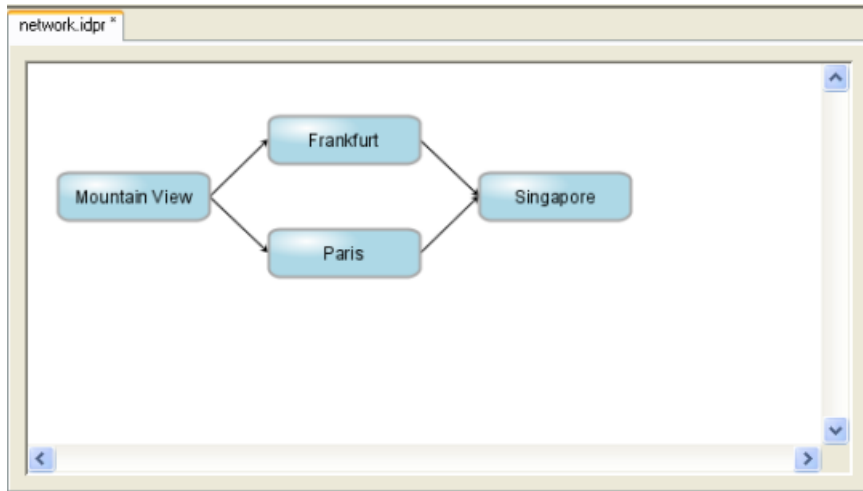
1. Click each `name` value.
2. Rename the nodes to Paris, MountainView, Frankfurt, and Singapore.
3. Click the **Add Column** button.
4. Add a column called `status`, one called `minor`, and one called `major`.
5. Click in the new columns to enter the data values shown in the following figure.





*A simple network model in memory*

6. Once you have defined your data model and data values, click Next to select the type of diagram.
7. Select the X/Y Diagram option.
8. Click Finish to exit the wizard.
9. Choose **File>Save As** and save the network diagram as `network.idpr`.
10. Keep this diagram open to use in the next section.



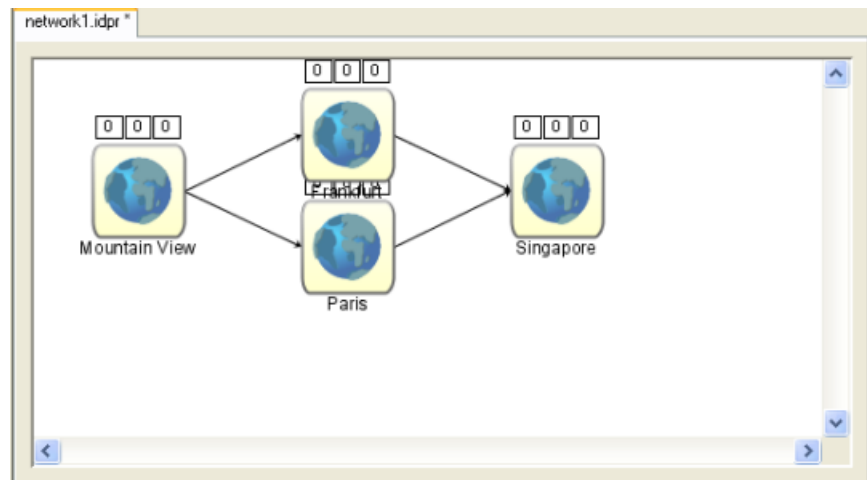
*The initial network diagram*

## Changing the graphic representation of the nodes

You will change the default graphic representation of the nodes in the initial network diagram.

1. Select the node rule in the Style Rules tree.
2. Click the Assign symbol button at the bottom of the Styling Customizer.
3. From the Shared Symbols palette, select the NetworkDevice symbol in the Network category.
4. Click Apply.
5. In the Styling Customizer, specify the value @name for the name parameter.
6. Press ENTER.

See the following figure.



*NetworkDevice symbol applied to the diagram*

7. Choose **File>Save As** and save the network diagram as `network1.idpr`.
8. Keep this diagram open to use in the next section.

---

## Adding decorations for alarm severities

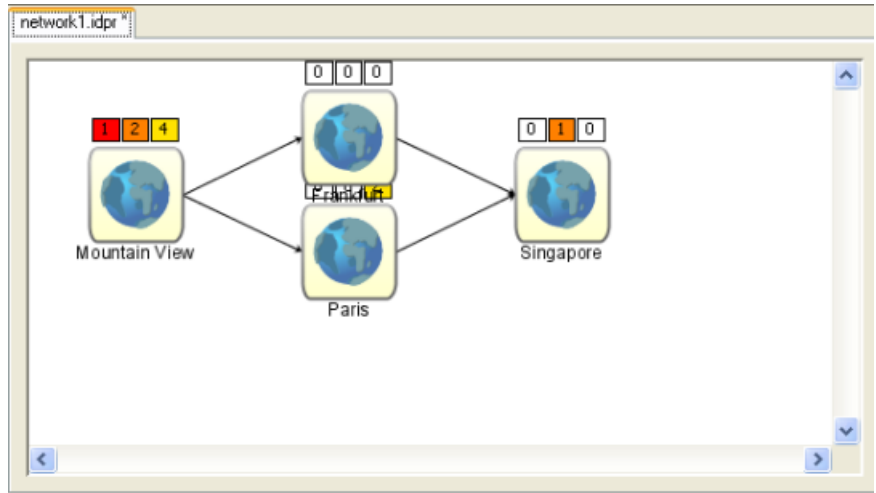
You can customize nodes represented by symbols to show changes in the objects in your diagram, for example changes in the status of network nodes. A symbol gives you scope for complex representations and allows you to modify the styling dynamically at run time.

### To add decorations to the nodes:

1. Select the node rule in the Style Rules tree and choose **Edit>Create Style Rule**.
  - a. In the Conditions page, specify the condition `status equals Unavailable`.
  - b. Click Finish.
  - c. In the Style Rules tree, rename the new rule to `node[status=Unavailable]`.
  - d. With the `node[status=Unavailable]` rule selected, set the `criticalAlarms` parameter value to 1.
  - e. Press ENTER.
2. Select the node rule in the Style Rules tree and choose **Edit>Create Style Rule**.
  - a. In the Conditions page, specify the condition `major exists and not null`.
  - b. Click Finish.
  - c. In the Style Rules tree, rename the new rule to `node[major]`.
  - d. With the `node[major]` rule selected, right-click the field `majorAlarms` in the Styling Customizer.
  - e. Choose Enter an Expression.
  - f. Delete the current value and specify `@major`.
  - g. Press ENTER.
3. Select the node rule in the Style Rules tree and choose **Edit>Create Style Rule**.
  - a. In the Conditions page, specify the condition `minor exists and not null`.
  - b. Click Finish.
  - c. In the Style Rules tree, rename the new rule to `node[minor]`.
  - d. With the `node[minor]` rule selected, right-click the field `minorAlarms` in the Styling Customizer.
  - e. Choose Enter an Expression.
  - f. Delete the current value and specify `@minor`.
  - g. Press ENTER.

As you add rules, you see the diagram change to reflect the current status of the network.

The resulting diagram should look like the following figure:



*Network diagram with alarm decorations*

4. Save the network diagram and keep it open to use in the next section.

If your diagram is connected to network data that changes dynamically, the icons will reflect the availability of nodes and any alarm recorded on each node.


**To test this behavior:**

1. Double-click a node to display its data properties.
2. Change the alarm or status value.
3. Watch the diagram change.

## Adding a node to a diagram

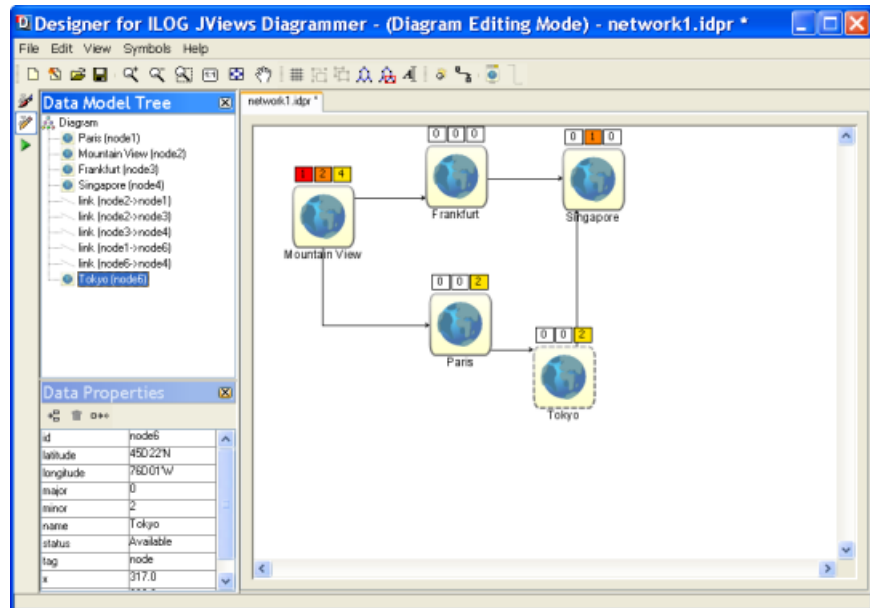
This section explains how to add a node to the network diagram, available as project file `network1.idpr`.

1. If you do not still have the Network example open, open it by choosing **File>Open** and browsing to the `network1.idpr` file.
2. Switch to the Diagram Editing Mode by clicking the middle icon in the vertical toolbar.

3. Click the node icon in the horizontal toolbar .
4. Click the link between the Paris and Singapore nodes.


If you click on top of an existing link, the Designer splits the link and inserts the new node.

5. Move the new node away to show the links.  
The link data has been modified appropriately by the Designer.
6. Modify the data of the new node as shown in the Data Properties panel to set the correct name (Tokyo).
7. Enable the link layout by choosing **Edit>Options** in Style Editing mode and checking the corresponding option.

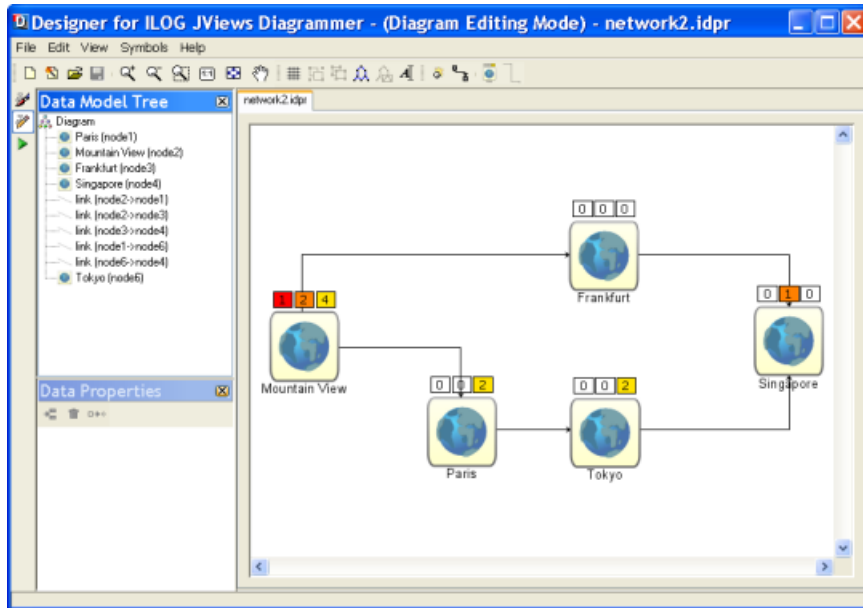


*Inserting a node on an existing link*

You can perform a layout operation to improve the appearance.

For example, the following figure shows the network diagram in figure *Inserting a node on an existing link* after a Layout All Nodes operation, .

8. Choose **File>Save As** to save the revised version and enter the name `network2`.



*Laying out all nodes after adding a new one*





# *Using subgraphs*

Explains how to create a group, which is a node that contains a subgraph of other nodes.

## **In this section**

### **Creating subgraphs by grouping objects**

Explains how to create a subgraph.


### **Styling subgraphs**

Explains how to customize the names of the subgraphs.

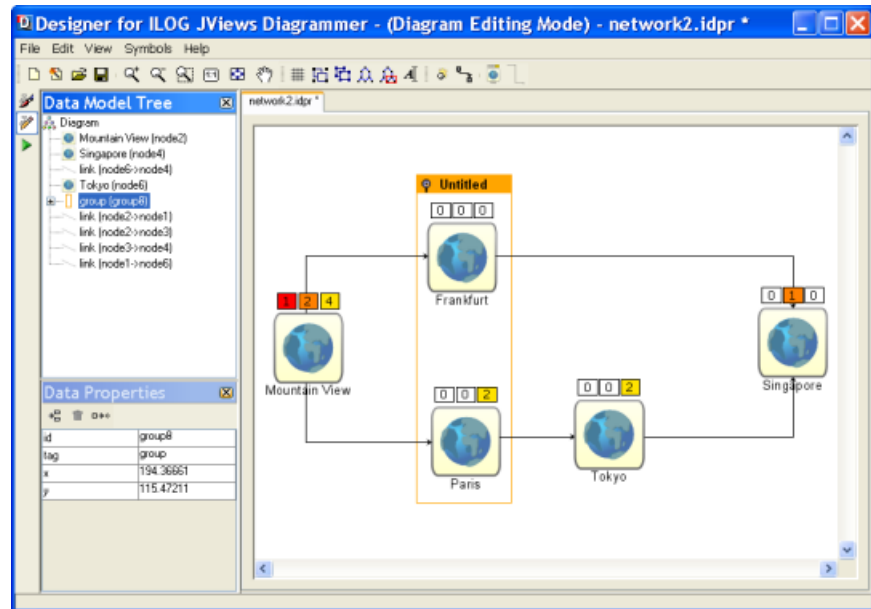
## Creating subgraphs by grouping objects

Subgraphs are useful to represent different levels of a hierarchy in a single diagram and to view each level separately. As an example, you can have the following levels in a network diagram: regions, nodes, and equipments.

### To create subgraphs:

1. Open the network example which you laid out in the previous section by choosing **File>Open** and selecting the project file `network2.idpr` if it is not still open.  
  
In this example, there is only the node level. To show the use of subgraphs, you can create a regional level.
2. Check that subgraphs are enabled by choosing **Edit>Options** in Style Editing Mode and making sure that Subgraphs is selected.
3. Switch to Diagram Editing Mode by clicking the middle icon in the vertical toolbar.
4. Move the nodes so that Frankfurt is vertically above Paris.
5. Drag the mouse around these two nodes and click the Group Selection icon in the toolbar .

This creates a subgraph containing the group of nodes. On creation, the subgraph is selected and expanded, see the following figure.



*An expanded group containing nodes*

6. Select Singapore.

7. Shift-click to add Tokyo to the selection.

Do not drag to select this time as that would include the link between these two nodes in the selection.

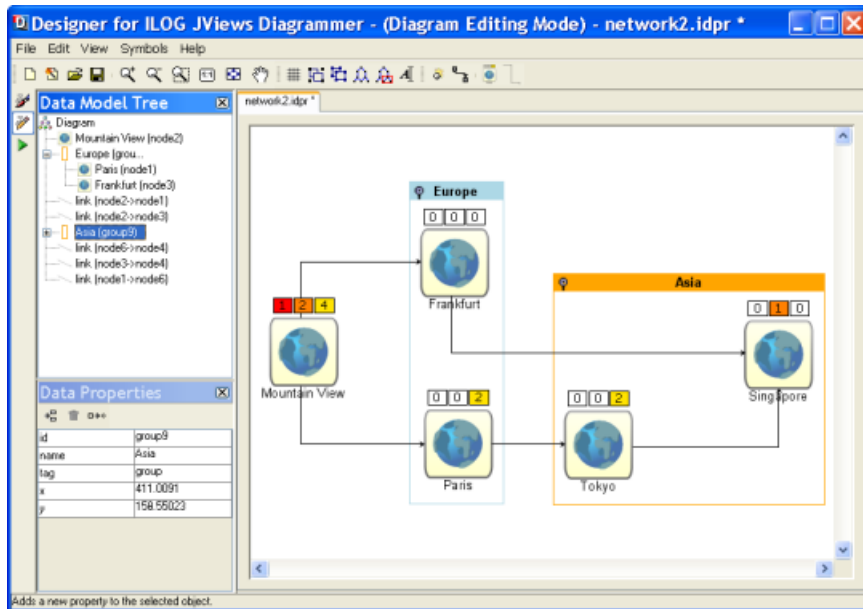
8. Click the Group Selection icon in the toolbar to create a second subgraph.

## Styling subgraphs

Now you will give the two groups meaningful names to identify them.

### To identify the two groups of nodes:

1. Select the first group.
  - a. Under Data Properties, click the New Property icon.
  - b. Enter name.
  - c. Set the value of name to Europe.
  - d. Press ENTER.
2. Select the second group.
  - a. Under Data Properties, click the New Property icon.
  - b. Enter name.
  - c. Set the value of name to Asia.
  - d. Press Enter.



*A new property for the group title*

The result is to set the title of the group to the value of name instead of Untitled.

### To style the representation of the subgraphs:

- ◆ Switch back to Style Editing Mode by clicking the top icon in the vertical toolbar.

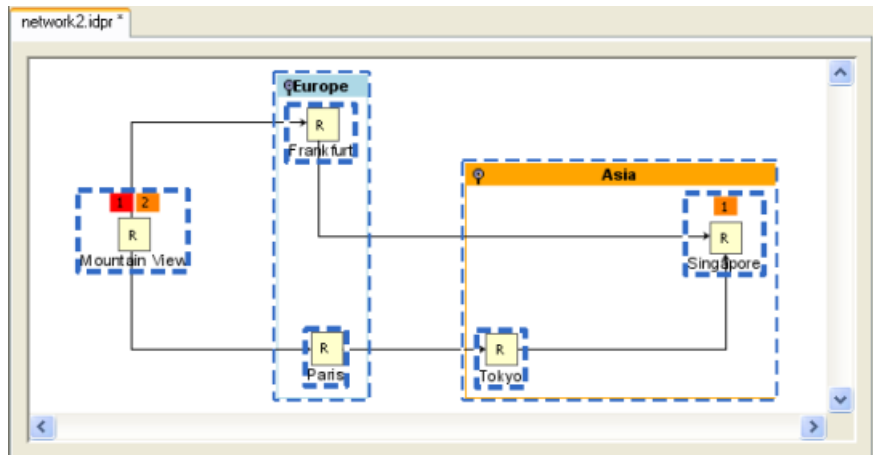
For further customization purposes, note that the user-defined type (`tag` property) of a group is `group` and that the state of a group can be `expanded` or `collapsed` as well as `selected` or `legend`.

## Displaying different levels of detail

You can define three different levels of detail for displaying the nodes of your diagram: high, medium, and low. To do so, you need to create the corresponding rules and assign a different symbol to each rule.

### To define the rules and graphic representation for the three levels of detail:

1. If you do not have the Network example open, open it by choosing **File>Open** and browsing to the `network2.idpr` file.
2. Select the `node` rule in the Style Rules tree and choose **Edit>Create Style Rule**.
  - a. In the Conditions page, specify the condition `is medium detail`.
  - b. Click Finish.
  - c. With the `is medium detail` rule selected, click the Assign symbol button in the Styling Customizer.
  - d. Select the `NetworkDeviceSmall` symbol in the Shared Symbols palette.
  - e. Click Apply.

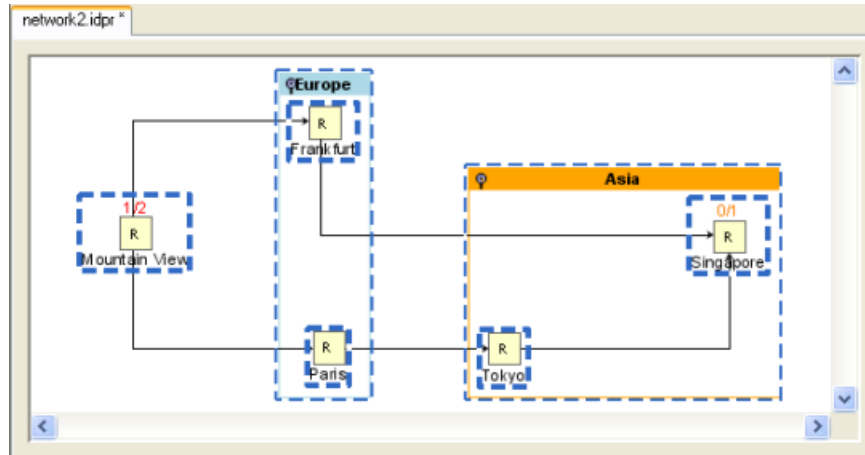


*Diagram display in medium level of detail*

3. Select the `node` rule in the Style Rules tree again and choose **Edit>Create Style Rule**.
  - a. In the Conditions page, specify the condition `is low detail`.
  - b. Click Finish.
  - c. With the `is low detail` rule selected, click the Assign symbol button in the Styling Customizer.
  - d. Select the `NetworkDeviceTiny` symbol in the Shared Symbols palette.
  - e. Click Apply.

4. For each of these rules, click the triangles in front of the parameters `name`, `criticalAlarms`, `majorAlarms`, and `minorAlarms` so that they point upwards.

By doing so, the level of detail rules will inherit the value of these parameters in the rules in which they have been defined.



*Diagram display in low level of detail*

The symbols `NetworkDeviceSmall` and `NetworkDeviceTiny` are simplified versions of the `NetworkDevice` symbol. They have the same parameters as the `NetworkDevice` symbol but are displayed with less detail.

The third level of detail, `high`, is the default detail level at which the nodes are displayed and, as such, does not need to be defined with a rule.

**To display the diagram in different levels of detail once these rules have been defined:**

- ◆ Use the **View>Level of detail** menu items.

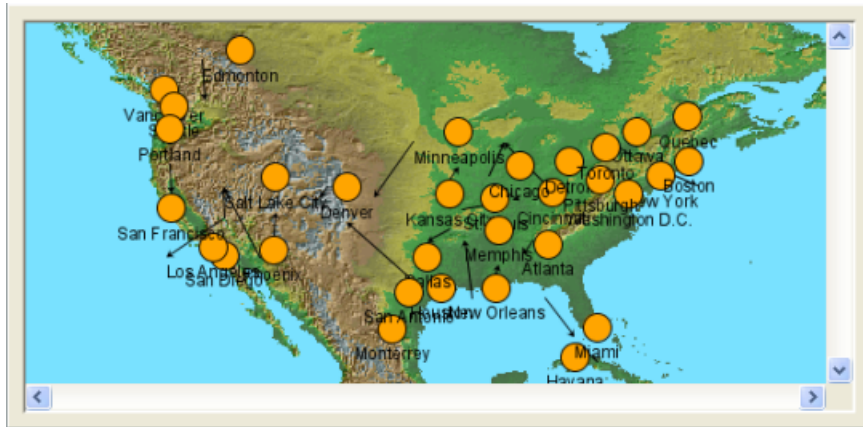
-- or --

- ◆ Click the corresponding rules in the Style Rules tree.

## Using background maps

A background map is typical of applications that require georeferenced data, for example, vehicle routing or network management. To make use of predefined maps supplied with JViews Diagrammer or to create your own maps with IBM® ILOG® JViews Maps, you must have purchased and installed the JViews Maps product. The JViews Map Builder and JViews Maps API of JViews Maps allow you to create maps for a specific region with features such as roads and boundaries.

The New Diagram Wizard allows you to select a Map Diagram, see the following figure.



*A diagram with a background map*

Note that you can see this diagram if you load the XML data file supplied as `<installdir>/jviews-diagrammer86/bin/designer/data/examples/example.xml`; the New Diagram Wizard sets the Map Diagram type by default because of the latitude and longitude values present in the data.

Note also that users cannot interact with the map: it is a static file.

If you do not select a Map Diagram in the New Diagram Wizard, you can later add a map to an existing diagram.

### **To add a background map:**

1. Choose **Edit>Options**.
2. Check the Background Map option.
3. Click OK.

The default map in the Designer is a world map. Your nodes are positioned according to their latitude and longitude attributes. If the data model does not include these attributes, the nodes are positioned in the top left corner of the map.

### **To change the background map:**

1. Select Background Map in the Options section of the Style Rules.
2. Set Map to the appropriate file name in the Styling Customizer.



# *Importing data from a database*

Explains how to load data from a database and map it.

## **In this section**

### **Loading the database**

Explains how to import the JDBC database.

### **Connecting to the database**

Explains how to configure the connection to the JDBC database.

### **Mapping the database**

Explains how to associate the data with properties.

### **Editing the data model**

Explains how to edit the database data model.

### **Selecting the diagram type**

Explains how to select a diagram type in the New Diagram wizard.

### **Choosing a Graph Layout Theme**

Explains how to specify a graph layout.

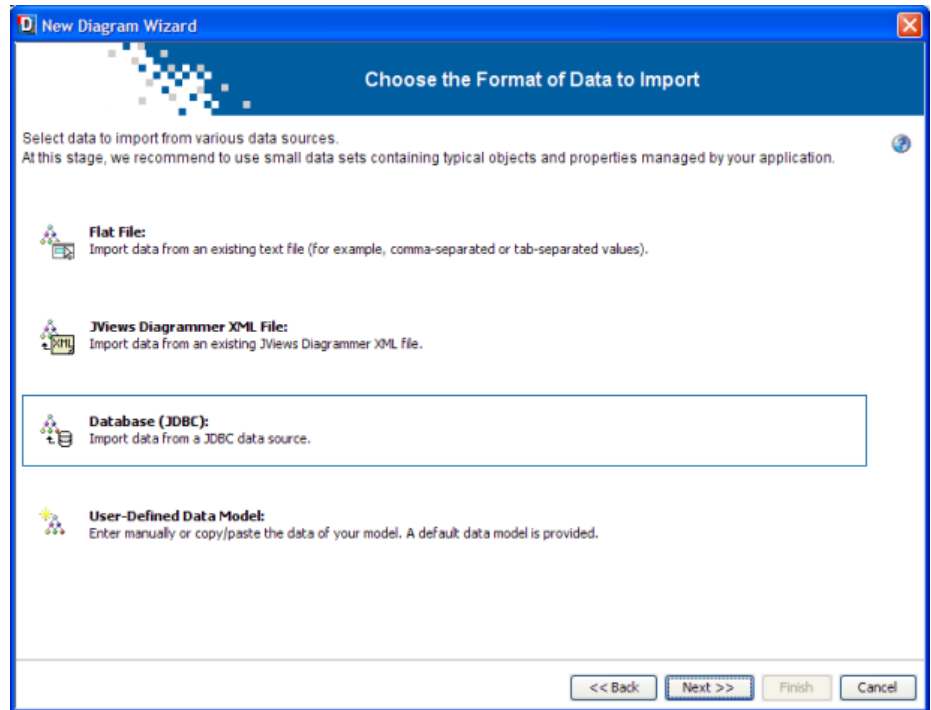
## Loading the database

The JDBC data source allows you to import data from any database supported by Java DataBase Connectivity.

### To import data from a JDBC database:

1. Choose **File>New from Wizard**.
2. Click Next.
3. Choose the database option.

See the following figure.



4. Click Next to move to the Import Data from a JDBC Data Source page.

---

## Connecting to the database

The Import Data from a JDBC Data Source page allows you to define a database connection configuration for nodes and links and then load your data.

This section makes use of the Arizona example.

### To create a database connection configuration:

1. Click the New Configuration button to name a new database connection configuration.
2. Name it Arizona.
3. Specify a URL to connect to the database.

The Universal Resource Locator (URL) can indicate a connection to a remote database or a path to a local file. It specifies the driver to use and the path to the file or database.

The default driver is Microsoft® Excel® ; this is correct for the Arizona data. The path is relative to the Designer run file location.

- a. Click the Ellipsis button.
- b. Browse to select the data file `ArizonaData.xls`.

Note that the Designer replaces only the file name in the URL.

4. Specify your user name and password, if these security items are required

If your data is in a local file (as the Arizona data is), you will not need to enter a user name and password. If your data is in a database, the Database Administrator can provide the information.

5. Specify an SQL query string for nodes (in the Nodes tab) and an SQL query string for links (in the Links tab).

If the database is a Microsoft® Excel® file, you specify the name of the sheet instead of a table name. The default sheet names in Excel® are just Sheet1, Sheet2, and so on.

The default query shown selects all columns from an Excel® sheet called Nodes, as follows:

```
select * from [Nodes$]
```

In the example supplied as `ArizonaData.xls`, suitable queries are as follows:

For node data:

```
select T1.*, T2.Availability, T2.Throughput from [Sheet1$] as T1, [Sheet2$]
as
T2 where T1.City = T2.City
```

For link data:

```
select * from [Sheet3$]
```

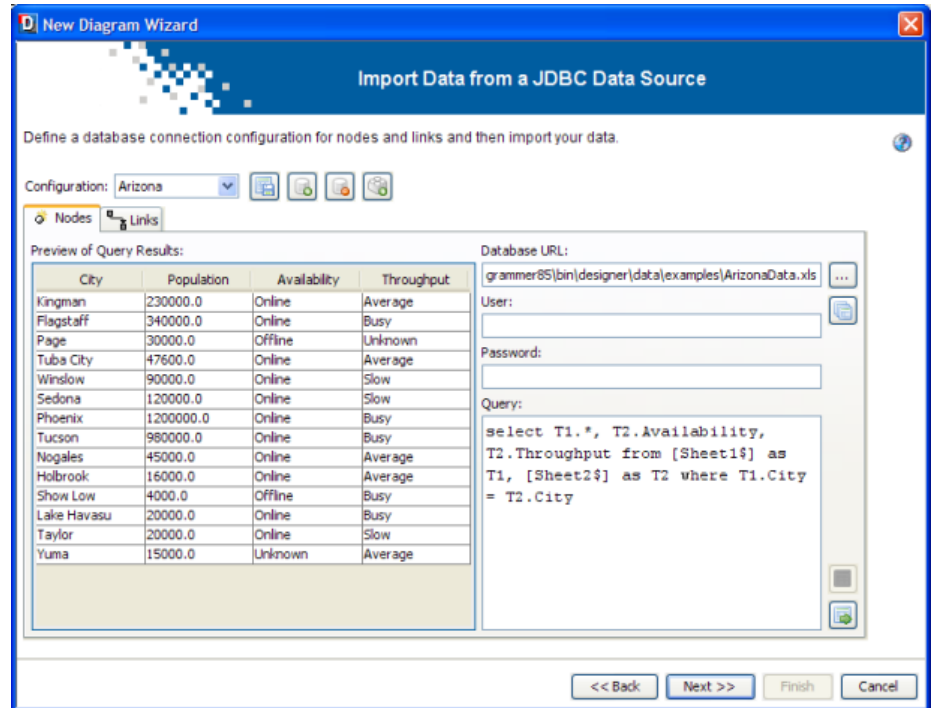
6. Click the Execute Query button to execute the SQL query.

7. Click the Save Configuration button to store the database connection configuration for future use.

The database connection configuration is stored in the project file.

Note that the IBM® ILOG® predefined connections cannot be saved. In the case of a predefined connection, the Save Configuration button is disabled.

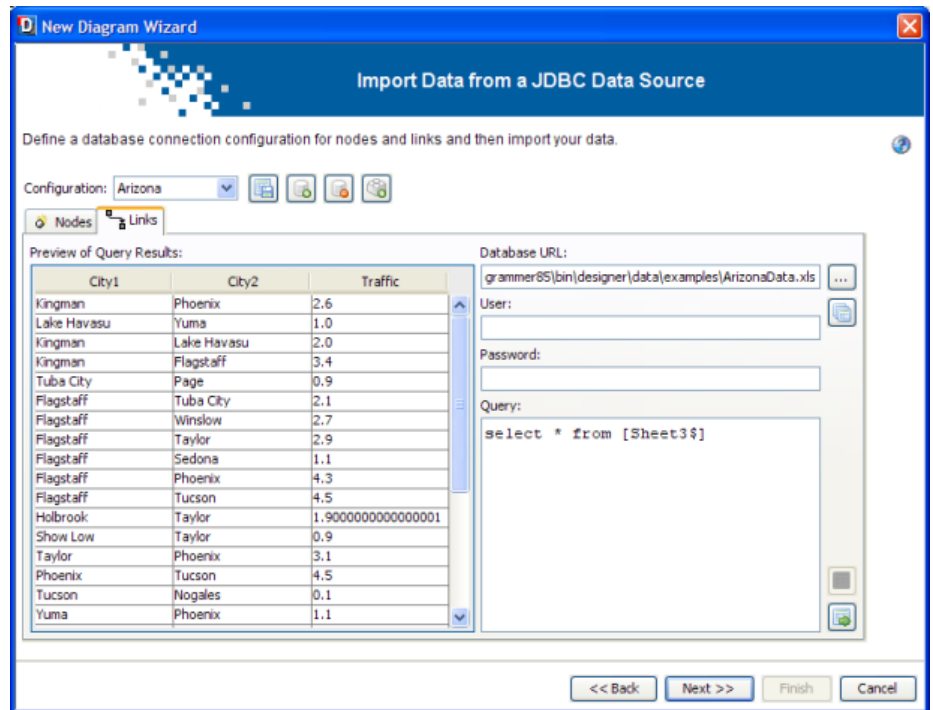
After executing an SQL query, you see the data in tabular form in the Preview pane, see the following figure for node data.



*Retrieving Arizona node data*

8. Check that the data retrieved is as required.

See also the following figure for link data.



*Retrieving Arizona link data*

9. Click Next to proceed.

---


## Mapping the database


The Mapping page allows you to specify the mapping of the database columns to graphic properties for nodes and for links.

Only mapped properties will be part of the data model.

Properties that are mandatory must be mapped; if they are not mapped, you will not be able to move forward in the wizard.

Optional properties do not need to be mapped, but if they are not mapped, they will not be in the final data model.

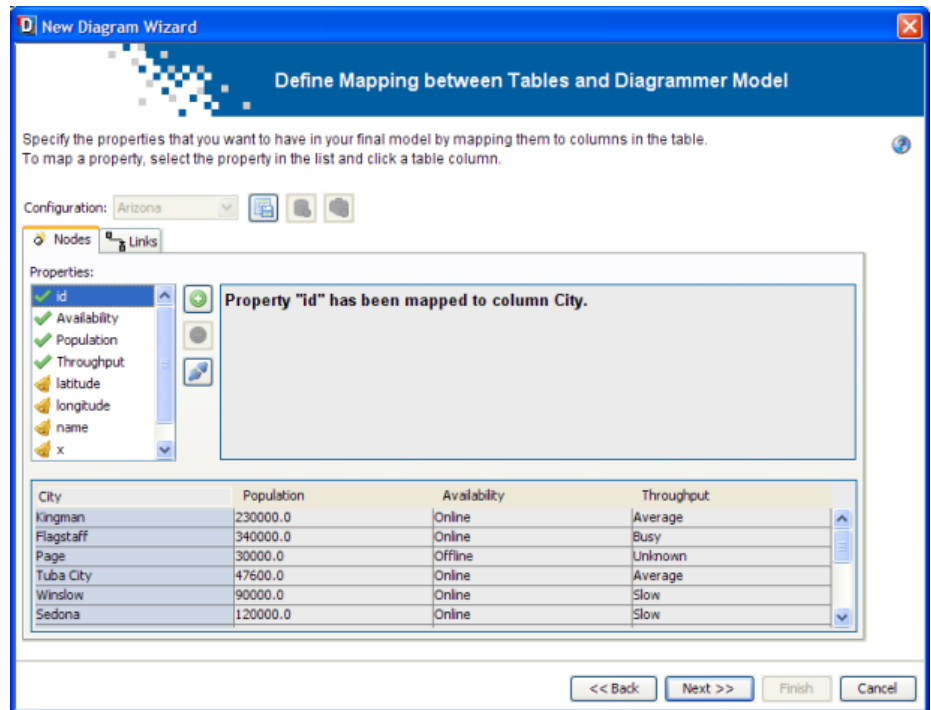
Properties that are not mapped to node data and require to be mapped appear with the following status  in the list of properties, and the same status appears on the Nodes tab.

Optional properties that are not mapped to node data appear with status  in the list of properties.

### To define the mapping for nodes:

1. In the Nodes tab, click the property `id`, then click the column `City`.

Once the `id` is mapped, the preview shows the nodes separately, see the following figure.



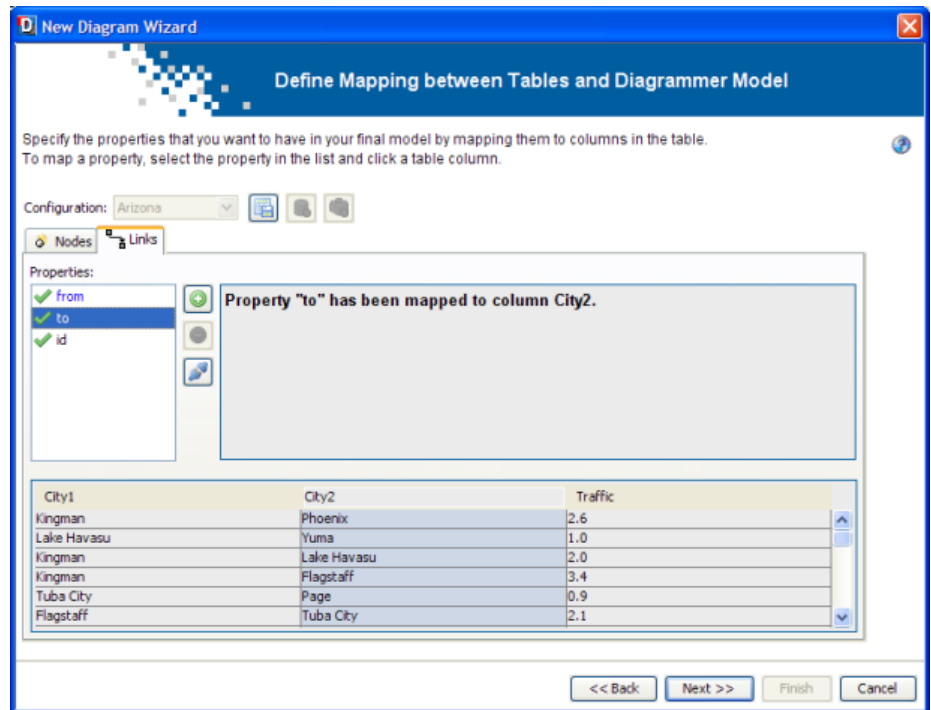
*The mapping for nodes*

2. Click the diskette icon to save the mapping.

**To define the mapping for links in the Links tab:**

1. Click the property `from`, then click the column `City1`.
2. Click the property `to`, then click the column `City2`.

Once the `from` and `to` nodes are mapped, the preview shows the nodes and links, see the following figure.




*The mapping for links*

3. Click the diskette icon to save the mapping.


Some commonly required properties are available in the list (such as x and y). If you need to, you can add further properties.

**To map extra properties:**


1. Click the  button to add a property.
2. With the new property selected, map it by selecting a column.

You can remove unwanted properties.

**To remove the mapping of a property:**

1. Select the property in the list.
2. Click the  button.

**To remove a property:**

1. Select the property in the list.
2. Click the  button.



Once you have the correct mapping for both nodes and links and have saved it, click Next.

---

## Editing the data model

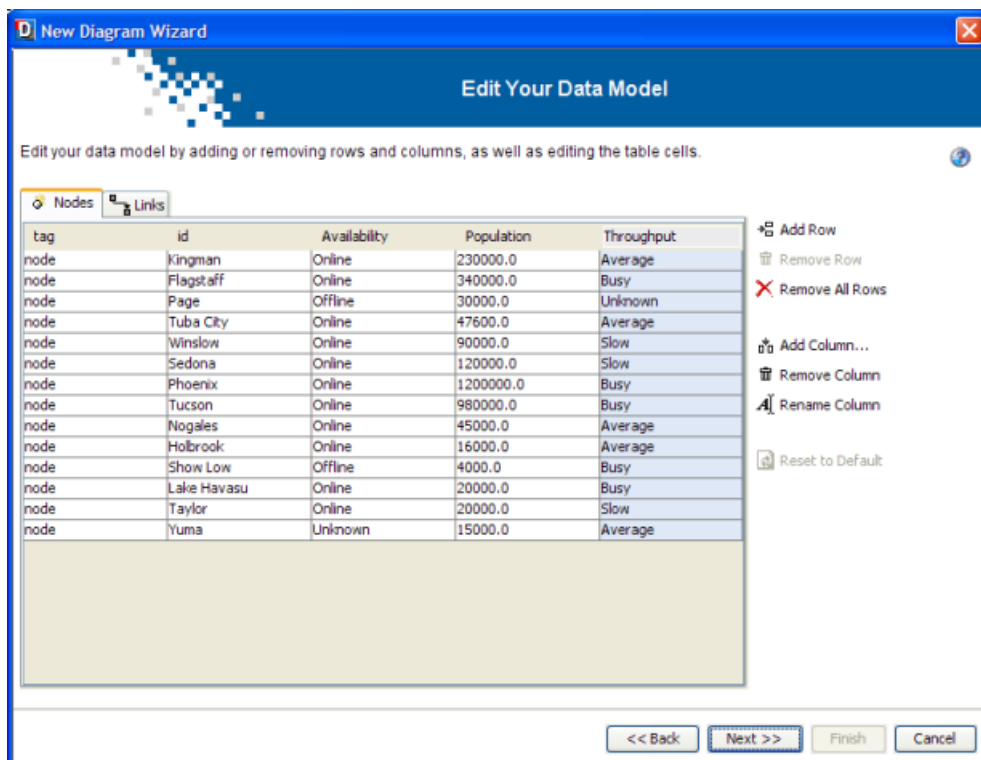
You can edit the basic data model to:

- ◆ Add and remove rows (a row represents an object in the data model)
- ◆ Add, remove, rename columns (a column represents a property of the objects in the data model)
- ◆ Change cell values (a cell represents a property of a specific object)
- ◆ Remove all objects
- ◆ Restore the data model to its initial state
- ◆ Paste data from the clipboard

If you click **Remove All Rows**, you clear all the data values. To retrieve the values you saw initially, click **the Reset to Default** button.

Properties `id` and `tag` are mandatory for the SDM model and therefore cannot be removed or renamed by the user.

**Paste from Clipboard** replaces the contents of the current table by the information present in the clipboard. It asks if the user wants the column names to be named based on the first row or not. If so, the first line of data is used to give the column names. If not, the column names are named as A, B, C, and so on, and can later on be renamed by the user manually.



*Editing the database data model*

You will keep this data model as it is.

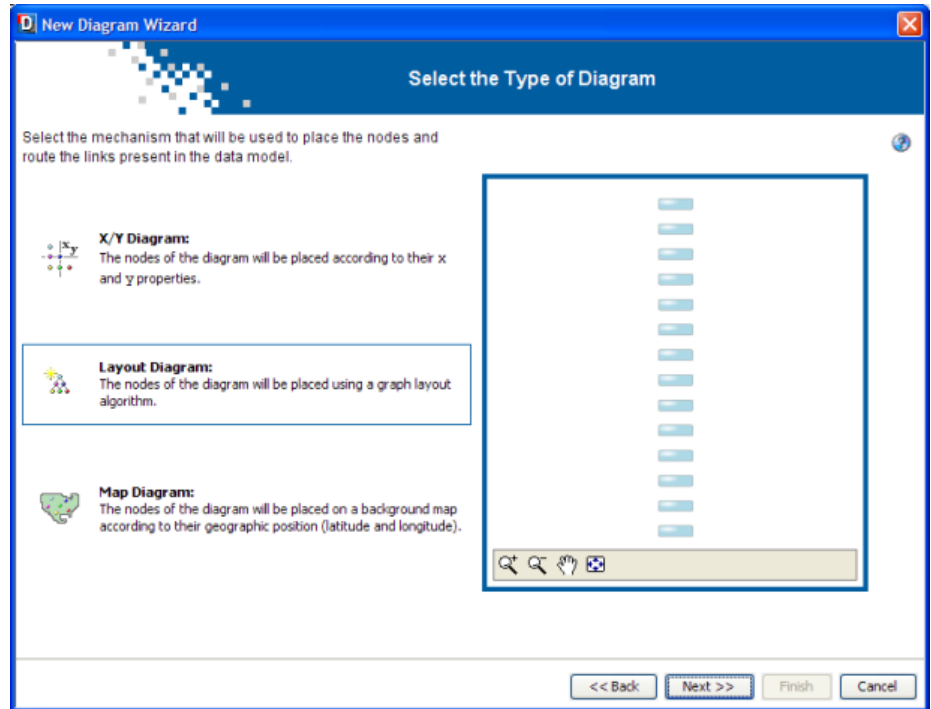
**To validate your data model:**

- ◆ Click Next.

## Selecting the diagram type

You need to select a diagram type for the correct positioning of the nodes.

1. Select the Layout Diagram, see the following figure.



*Specifying a layout diagram*

**Note:** You can only create a Map Diagram if you have purchased and installed the IBM® ILOG® JViews Maps product. Otherwise, a warning message appears.

2. Click Next.

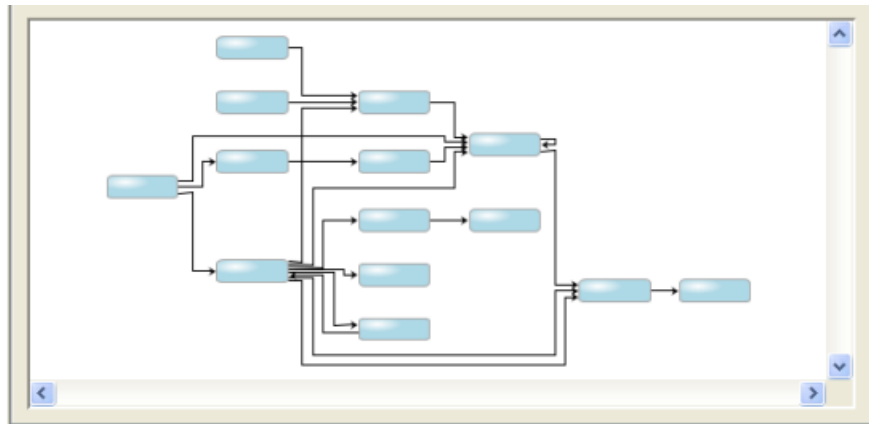
---

## Choosing a Graph Layout Theme

The default layout is acceptable.

1. Click Next.
2. If your wizard session has been successful, click Finish.  
-- or --
3. If you need to make changes, click Back to return to specific wizard pages.

The resulting diagram is as shown in the following figure.



*The initial Arizona Logical diagram*

This diagram is available as is in `<installdir>/jviews-diagrammer86/bin/designer/data/examples/ArizonaLogical.idpr`. You can style it to show the Population, Availability, and Throughput data for nodes and the Traffic data for links in appropriate ways, see *Using data model expressions*.

4. When the diagram is as you wish it to be, choose **File>Save As** and enter a new name.



# *Using data model expressions*

Explains an advanced feature for inserting references to the data model or expressions in certain property values.

## **In this section**

### **About references and expressions**

Explains what references and expressions are.

### **Using expressions in text properties**

Explains how to define a text property using an expression or reference to the data model.

### **Using expressions in numeric or fixed-value properties**

Explains how to define a numeric or fixed-value property using an expression or reference to the data model.

### **Using Expression Assist**

Provides hints to use the Expression Assist facility to enter expressions.

### **Using variables**

Explains what are variables with regard to expressions and how to work with them.

### **Using custom functions**

Explains what are custom functions in relation to expressions and how to work with them.

### **Displaying help on expressions**

Explains how to obtain details about the format of expressions.

---

## About references and expressions

It is possible to use the values of data model attributes for styling. You can also use an expression involving one or more data property values for styling.

A reference to a model attribute is the name of the attribute preceded by the commercial at symbol, for example, @id.

An expression is signaled by the initial symbols @| for example, @|"name is: "+@id. Note that this example uses the concatenation operator, +, and that literal strings must be enclosed in quotation marks. A further example is @|@firstname+" "+@name, which is used in the genealogical tree example, see *Customizing genealogy nodes by type*. For full details of the expression syntax, see Expressions in *Developing with the JViews Diagrammer SDK*.

The way to enter references or expressions in the Designer depends on whether the styling property you want to set is a text property or not.

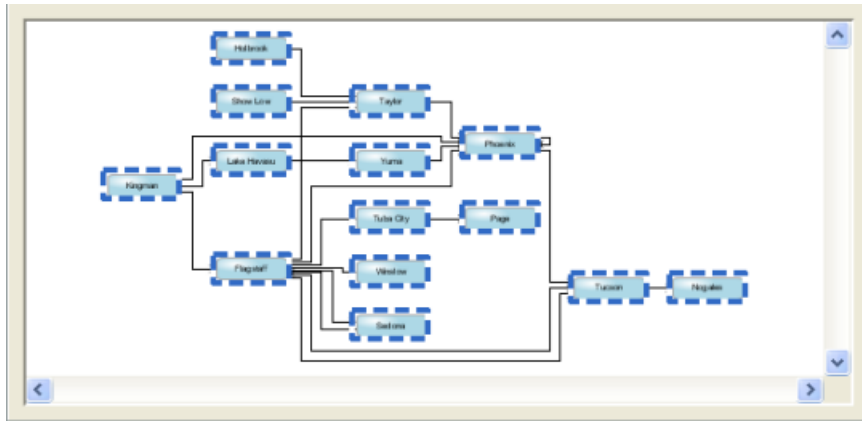


## Using expressions in text properties

In the Styling Customizer, for styling properties with a text value, you can directly enter references to model attributes or expressions containing references to model attributes.

1. If you do not have the `ArizonaLogical` example open, choose **File>Open** and browse to the `ArizonaLogical.idpr` file.
2. .
3. Select the node rule.
4. Change the value of the name parameter to `@id` in the Styling Customizer.
5. Press ENTER.

You see the names of the cities appear in the nodes, see the following figure.



*Data model attribute values in a diagram*

By default, name is set to `@name`, but in the Arizona data model, there is no attribute called name, instead the city names are mapped to the attribute called `id`.

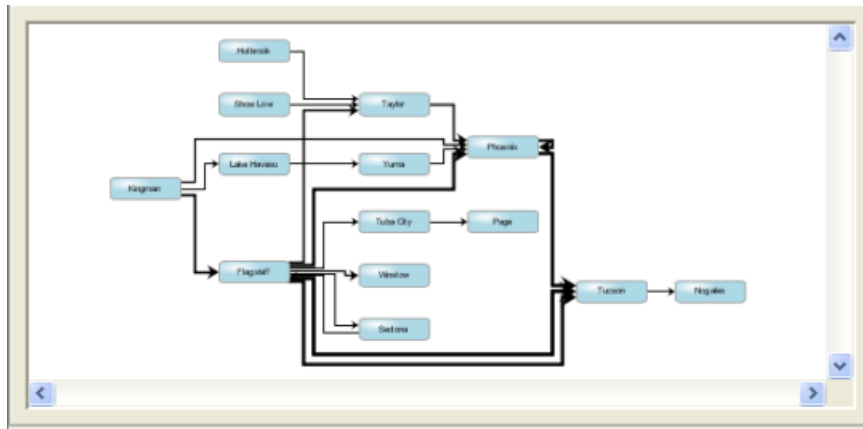
Note that if you create a new property in Diagram Editing Mode, it is a text property.

## Using expressions in numeric or fixed-value properties

For styling properties which usually have a fixed type of value (for example, numeric) or a fixed list of possible values (for example, `true/false`), you cannot enter a data model reference or expression directly. Instead, you must use the right-click menu.

1. In the `ArizonaLogical` example, select the `link` rule.
2. Select the `Width` tab in the `Styling Customizer`.  
The styling properties in this tab are numeric.
3. Increase the `Line width max.` to 50.
4. Click in the `Line width` field.
5. Clear the value.
6. Right-click and select `Enter an Expression`.
7. Enter `@Traffic`.
8. Press `ENTER`.
9. Click in the canvas to see the result better.

The line widths reflect the volume of traffic on the link, see the following figure.



*Numeric data model attribute value used in a diagram*

If the thickness of the lines is not sufficient, try entering `@|@Traffic*2` instead.

---

## Using Expression Assist

The Expression Assist is a utility feature that provides a quick way to enter expressions in the Expression text field by opening a scrollable list of available options.

Following are a few tips on how to use the Expression Assist:

- ◆ You can use the mouse or the keyboard to navigate and select a line in the list of options. Click or press ENTER to insert the selected expression into the text field.
- ◆ The list of options opens when the cursor is placed at a location where expression completion makes sense, for example, after a separator.
- ◆ The proposed expression options are syntactically correct, but may not be semantically correct.
- ◆ The list of expression options displays the expression options followed by a short description of the options.

**Example:**

1. Create a new diagram by choosing **File>New**.
2. In the diagram window, click one of the blue boxes.
3. In the Styling Customizer, at the right of the `label` property field, type `+`.

The Expression Assist displays the list of expression options. First in line are the model attributes, followed by the variables, then the functions.

---

## Using variables

Variables allow you to define sharable values like a color or a number. Variables can be used in an expression, where they will be substituted by their value at run time.

### To create or edit a variable:

1. Choose **Edit>Variables**.

The Variable definitions dialog opens with the list of currently defined variables.

- a. To change a variable, edit its name or value directly in the list.
- b. To remove a variable, select its row in the list and click Remove.
- c. To add a variable, click the Add button.  
A new row is created.
- d. Edit the name and value to set the variable.

2. Click Apply to accept the changes.

The Expression Assist will be automatically updated with the new variable set.

### To use a variable in an expression:

- ◆ Use the Expression Assist to select the right variable.

---

## Using custom functions

Custom functions can be registered to be used inside an expression.

### To register custom functions:

1. Choose **Edit>Functions**.

The Custom functions dialog opens with the list of currently registered custom functions.

- a. To remove a function, select it in the list and click Remove.
- b. To add a function, click Add and type the fully-qualified Java™ class name in the text field.

The new function will be added to the list.

If the function is valid, its name and description can be seen in the details area.

To be valid, a custom function must meet the following two criteria:

- ◆ It must be a Java class implementing `ilog.views.sdm.renderer.ILvSDMCSSFunction`.
- ◆ It must be already in the CLASSPATH of the Designer.

If the function name cannot be loaded into the JVM™, a red warning is displayed in place of the name and short description.

2. To accept the changes, click Apply.

Note that only the valid functions will be registered and shown by the Expression Assist. The others will be removed.

### Example:

1. Create a new diagram by choosing **File>New**.
2. Choose **Edit>Functions**.
3. Click Add.
4. Type `ilog.views.sdm.renderer.ILvSDMCSSFunction$Test` in the Add new function dialog.
5. Click OK.

The new function appears in the list.

The function is only for test. It returns its first argument.
6. Click Apply.
7. In the diagram window, click one of the blue boxes.
8. In the Styling Customizer, at the right of the `name` property field, type `+`.

The Expression Assist displays a list of expression options.
9. Select `testFunction`.

The function is added in the expression.

10. Inside the brackets, type :up.
11. Press ENTER to validate the expression.

All nodes have now :up appended to their label, for example Node 1:up.

---

## Displaying help on expressions

You can display a help window about expressions.

1. Right-click a property field of the Styling Customizer.
2. Select Help about expression in the shortcut menu.

This help explains the format of expressions, lists the data model properties that you can use, and gives some examples of expressions.





# *Writing an application*

Explains how to integrate the diagram component of the genealogy example into a Swing application.

## **In this section**

### **Deployment considerations**

Indicates the deployment paths and functional types you have to take into account before building an application.

### **JViews Diagrammer classes available as beans**

Lists the JViews Diagrammer JavaBeans™ that you can use to build an application.

### **Creating a JViews Diagrammer application**

Describes the various ways to create a JViews Diagrammer application, from simple to more complex.

### **Printing the diagram from an `IlvDiagrammerApplication` instance**

Explains how to print a diagram from an application which is an instance of the `IlvDiagrammerApplication` class.

---

## Deployment considerations

Once you have specified your data source and the styling of the diagram component, you will need to integrate it into a Java™ application and deploy it. This phase includes the writing of some Java code.

In order to build and deploy an application containing a JViews Diagrammer diagram component, you need to first consider the required deployment path and the functional type of the application.

---

### Deployment paths

The deployment paths available are:

- ◆ Swing application
- ◆ JavaServer™ Faces (JSF)
- ◆ Thin client
- ◆ Applet

This section discusses the Swing application path. A ready-to-use Swing application based on a diagram component is supplied by the class `IlvDiagrammerApplication`.

For discussions of the JSF and thin client deployment paths, see *Using DHTML-based JSF components to build Web applications* and *Deploying an application as a DHTML-only thin client in Building Web applications*. For a description of the applet deployment path, see *Creating an applet in Developing with the JViews Diagrammer SDK*.

---

### Functional application types

The functional type of the application influences the choice of deployment path.

There are two functional types of application:

- ◆ Display and navigation, providing different views of data but no other interaction
- ◆ Editing, with a graphical editor that provides update capabilities

If you are building a display and navigation application, you can deploy it as a Swing application, a thin client or an applet.

If you are building a graphical editor, you can deploy it as a Swing application or possibly an applet if the user does not need to save a changed version.

---

## JViews Diagrammer classes available as beans

The Designer allows you to create a diagram that can be loaded in the main component of JViews Diagrammer, the `IlvDiagrammer` component. This component is a JavaBean™ that you will integrate into the Swing GUI of your application. Further JavaBeans are provided to build up a GUI in your favorite IDE. These JavaBeans supply commonly required components such as menus and toolbars.

The full list of JViews Diagrammer beans is shown in alphabetical order in the following table.

### ***JViews Diagrammer JavaBeans***

<code>IlvDiagrammer</code>	A diagram component; it wraps an SDM view and an SDM engine. There can be one or more in an application.
<code>IlvDiagrammerEditBar</code>	A predefined toolbar containing actions for editing the objects in a diagram component.
<code>IlvDiagrammerEditMenu</code>	A predefined Edit menu for use in an application based on a diagram component. The default actions are: Undo, Redo, Cut, Copy, Paste,

	Duplicate, Select All, Delete, Group, Ungroup, Layout All Nodes, Layout Selected Nodes.
<code>IlvDiagrammerFileMenu</code>	A predefined File menu for use in an application based on a diagram component.
<code>IlvDiagrammerHelpMenu</code>	A predefined Help menu for use in an application based on a diagram component.
<code>IlvDiagrammerMenu</code>	A predefined menu for use in an application based on a JViews Diagrammer diagram component.
<code>IlvDiagrammerMenuBar</code>	A predefined menu bar for use in an application based on a JViews Diagrammer diagram component. The default menus are File, Edit, View, Options, Help.
<code>IlvDiagrammerOptionsMenu</code>	A predefined Options menu for use in an application based on a JViews Diagrammer diagram component. The default actions are: Editing, Sticky Modes, Auto Edit Label.
<code>IlvDiagrammerOverview</code>	An overview component showing the whole extent of the diagram and any background map in a reduced view.
<code>IlvDiagrammerPaletteBar</code>	A predefined toolbar containing actions for creating new nodes and links in an application based on a JViews Diagrammer diagram component.
<code>IlvDiagrammerPropertySheet</code>	A component to view and edit the properties of the selected object in a JViews Diagrammer diagram component.
<code>IlvDiagrammerStatusBar</code>	A predefined status bar for use in an application based on a JViews Diagrammer diagram component
<code>IlvDiagrammerTable</code>	A component to display the data model of a JViews Diagrammer diagram component as a table.
<code>IlvDiagrammerToolbar</code>	A toolbar in which the contents are set through an actions property. Predefined action arrays are provided, for example, <code>setActions(editActions)</code> and <code>setActions(viewActions)</code> .
<code>IlvDiagrammerTree</code>	A component to display the data model of a JViews Diagrammer diagram component as a tree.
<code>IlvDiagrammerViewBar</code>	A predefined toolbar containing actions for controlling the view in a JViews Diagrammer diagram component.
<code>IlvDiagrammerViewMenu</code>	A predefined View menu for use in an application based on a JViews Diagrammer diagram component. The default actions are: Zoom In, Zoom Out, Reset Zoom, Fit To Contents.

The ready-to-use Swing application supplied by the class `IlvDiagrammerApplication` makes use of all the JavaBeans listed.

The menu classes are derived from `IlvDiagrammerMenu` and are alternatives to using the standard `JMenu` class.

# ***Creating a JViews Diagrammer application***

Describes the various ways to create a JViews Diagrammer application, from simple to more complex.

## **In this section**

### **Creating an application instance**

Explains how to create an instance of `IlvDiagrammerApplication`.

### **Customizing the predefined application**

Describes which parts of the predefined `IlvDiagrammerApplication` you can customize.

### **Creating a basic application from JViews Diagrammer beans**

Explains how to create an application by combining JavaBeans™ .

### **Adding further components**

Explains how to create more complex applications.

### **An editing application created from JViews Diagrammer beans**

Describes an example of a more complex application provided with JViews Diagrammer.

---

## Creating an application instance

The Designer is used with sample data that contains only enough information to specify the drawing style.

The class `IlvDiagrammerApplication` is a ready-to-use application to display or edit diagrams in diagram components (instances of `IlvDiagrammer`).

To create an instance of an application, the `CLASSPATH` environment variable must contain the JViews Diagrammer JAR files, as well as the palette JAR files that contain the symbols used in the projects.

### To create an `IlvDiagrammerApplication` instance:

- ◆ Launch the application from the command line, passing it the style sheet created in the Designer as the `style` argument, as in the following example.

```
java ilog.views.diagrammer.application.IlvDiagrammerApplication -style
example.css
```

You can pass further command-line arguments to control which components are visible, to allow editing of the diagram, and so on. See the reference documentation for `IlvDiagrammerApplication` for a list of the possible command-line arguments.

### To modify the data source of the application in order to display your own data:

1. Retrieve the current diagrammer,

```
IlvDiagrammerApplication.getCurrentDiagrammer()
```

2. Modify its data source as explained in the section 'Specifying a different data source' of *Next steps after the Designer*.

---

## Customizing the predefined application

You can customize the ready-to-use application by subclassing `IlvDiagrammerApplication`. In the subclass, you can customize the following aspects of the application:

- ◆ Working area: can be single-document (default) or multidocument with `JDesktopPane` and internal frames
- ◆ Menu bar: default contents File, Edit, View, Options, Help
- ◆ Edit toolbar: by default horizontally above the working area
- ◆ View toolbar: by default vertically to the left of the working area
- ◆ Palette toolbar (optional)
- ◆ Tree showing data model (optional)
- ◆ Property Sheet (optional)
- ◆ Overview showing the whole of a (large) diagram (optional)

Runtime flags govern what is displayed at startup.

---

## Creating a basic application from JViews Diagrammer beans

An alternative to using the predefined `IlvDiagrammerApplication` class is to combine the JViews Diagrammer beans yourself. You can do this either using an Integrated Development Environment (IDE), or directly in your Java™ code.

**To create a very simple JViews Diagrammer application containing one diagram component ( `IlvDiagrammer` instance) and a view toolbar (`IlvDiagrammerViewBar` instance):**

- ◆ Write the following code

### A basic application that extends `JFrame`

```
/**
 * This is a very simple application based on a diagram component,
 *
 * This application displays a diagram component in a frame.
 * A simple data file is loaded and displayed using a
 * basic style sheet. A predefined toolbar is also used.
 */
public class BasicDiagrammerApplication extends JFrame
{
    // Create the main frame.
    //
    public BasicDiagrammerApplication()
    {
        // Set up the frame.
        //
        super("Basic Diagrammer Application");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocation(100, 100);
        setSize(400, 300);
        // Create the IlvDiagrammer instance.
        //
        IlvDiagrammer diagrammer = new IlvDiagrammer();
        // Allow selection.
        //
        diagrammer.setSelectMode(true);
        // Add the IlvDiagrammer instance.
        //
        getContentPane().setLayout(new BorderLayout());
        getContentPane().add(diagrammer, BorderLayout.CENTER);
        // Add a predefined toolbar for controlling the view
        // north of the frame.
        //
        getContentPane().add(new IlvDiagrammerViewBar(), BorderLayout.NORTH);
    }
    ;
    try {
        // Load a project file.
        //
        diagrammer.setDataFile(new URL("file:data/basic.idpr"));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



```
    }  
  }  
  /**  
   * The main method of the application.  
   */  
  public static void main(String[] args)  
  {  
    // Create an instance of the application, and  
    // show it.  
    //  
    new BasicDiagrammerApplication().setVisible(true);  
  }  
}
```

This sample is available at

**[file://<install\\_dir>/jviews-diagrammer86/codefragments/basic/src/html/BasicDiagrammerApplication.java.html](file://<install_dir>/jviews-diagrammer86/codefragments/basic/src/html/BasicDiagrammerApplication.java.html)**

---

## Adding further components

You can build more complex applications by adding further JViews Diagrammer beans to the Swing GUI of a basic application.

JViews Diagrammer beans are connected automatically to the currently active `IlvDiagrammer` instance of the GUI.

**All you need to do to add a bean is:**

1. Create it.
2. Add it to your Swing GUI.

No call is necessary to connect the bean to the `IlvDiagrammer` instance.

For example, you can make the basic application in *A basic application that extends `JFrame`* more complete by adding a menu bar to the frame, and an overview to the right of the panel, as shown in the following code example:

```
setJMenuBar(new IlvDiagrammerMenuBar());  
getContentPane().add(new IlvDiagrammerOverview(), BorderLayout.EAST);
```

---

## An editing application created from JViews Diagrammer beans

For an example of a more complex application, see the Editor sample, which is available at <mailto:<installdir>/jviews-diagrammer86/codefragments/editor/srhtml/DiagramEditor.java.html>.

This sample is a simple multidocument diagram editor based on the diagram component with the following:

- ◆ A horizontal toolbar for editing the diagram.
- ◆ Palette icons in the toolbar to create nodes and links in the diagram.
- ◆ A vertical toolbar to control the diagram view.

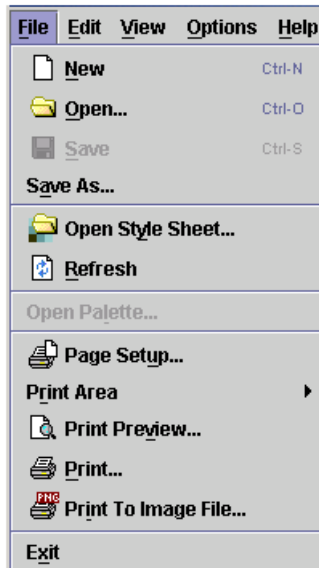
---

## Printing the diagram from an IlvDiagrammerApplication instance

When you use the `IlvDiagrammerApplication` class, you automatically have the same printing facilities for your diagram component(s) as you have in the Designer:

- ◆ Page setup (provided by the method `pageSetup`)
- ◆ Print a specified area (provided by the methods `setPrintArea`, `clearPrintArea`)
- ◆ Print preview (provided by the method `printPreview`)
- ◆ Printing one, all or selected pages (provided by the method `print`)
- ◆ Print to an image file (provided by the method `printToBitmap`; .jpg or .png format)

These operations are available from the **File** menu in the application, see the following figure.



**To print the diagram to the default printer from the application:**

- ◆ Click the Print button in the Print Preview window or in the main toolbar,
- or --
- ◆ Choose **File>Print**.

# *Data formats*

Describes the data formats that are supported by JViews Diagrammer and the Designer.

## **In this section**

### **About supported formats**

Indicates the most common data formats used in JViews Diagrammer files and the other supported formats.

### **Application data in XML files for JViews Diagrammer**

Describes in detail the elements used in the XML format.

### **Application data in flat files for JViews Diagrammer**

Describes in detail the elements of a flat file.

### **Application data in databases for JViews Diagrammer**

Describes in detail the elements of a database.

### **Project files**

Explains what a project file is and how it is implemented.

### **Style sheet files**

Explains how style sheets are implemented when you create a diagram.

---

## About supported formats

There are two main data formats used in the JViews Diagrammer files: XML (for application data and projects) and CSS for Java (for style sheets). JViews Diagrammer can also load application data from JDBC-compliant databases and the Designer can also load application data from flat files (for example, the CSV format).

If you have a custom data model, you should export the data to a project in Java™ code (see *Project files*) and then load the resulting XML file.

---

## Application data in XML files for JViews Diagrammer

You can import application data from an XML file provided that the XML elements conform to the diagram format.

The following code example shows the initial XML data file for the genealogy tree as loaded in *Getting started*.

### Simple genealogy tree data and model in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE diagram>

<diagram metadata="true" xmlns:sdm="http://www.ilog.com/JViews/SDM/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <person id="person_1">
    <property name="sex">male</property>
    <property name="sdm:y">264.91016</property>
    <property name="firstname">John</property>
    <property name="profession">teacher</property>
    <property name="deathdate">1960</property>
    <property name="sdm:x">35.429688</property>
    <property name="name">Smith</property>
    <property name="birthdate">1900</property>
  </person>
  <person id="person_2">
    <property name="sex">female</property>
    <property name="sdm:y">165.28516</property>
    <property name="firstname">Janet</property>
    <property name="deathdate">1970</property>
    <property name="sdm:x">35.429688</property>
    <property name="name">Whoop</property>
    <property name="birthdate">1903</property>
  </person>
  <couple id="couple_1">
    <property name="startdate">1920</property>
    <property name="sdm:y">215.09766</property>
    <property name="sdm:x">123.75</property>
    <property name="status">married</property>
  </couple>
  <couplelink islink="true" id="couple_11" from="person_1" to="couple_1" />
  <couplelink islink="true" id="couple_21" from="person_2" to="couple_1" />
  <person id="person_3">
    <property name="sex">male</property>
    <property name="sdm:y">310.3828</property>
    <property name="firstname">Andrew</property>
    <property name="profession">engineer</property>
    <property name="deathdate">1995</property>
    <property name="sdm:x">214.79688</property>
    <property name="name">Smith</property>
    <property name="birthdate">1922</property>
  </person>
```

```

<childlink islink="true" id="child_1_3" from="couple_1" to="person_3" />
<person id="person_4">
  <property name="sex">female</property>
  <property name="sdm:y">119.8125</property>
  <property name="firstname">Mary</property>
  <property name="profession">Advocat</property>
  <property name="sdm:x">214.79688</property>
  <property name="name">Smith</property>
  <property name="birthdate">1924</property>
</person>

```

```

<childlink islink="true" id="child_1_4" from="couple_1" to="person_4" />
<person id="person_5">
  <property name="sex">male</property>
  <property name="sdm:y">212.97656</property>
  <property name="firstname">Richard</property>
  <property name="profession">Writer</property>
  <property name="deathdate">1995</property>
  <property name="sdm:x">214.79688</property>
  <property name="name">Bilbot</property>
  <property name="birthdate">1920</property>
</person>
<couple id="couple_2">
  <property name="startdate">1942</property>
  <property name="sdm:y">166.39453</property>
  <property name="enddate">1960</property>
  <property name="sdm:x">305.84375</property>
  <property name="status">divorced</property>
</couple>
<couplelink islink="true" id="couple_42" from="person_4" to="couple_2" />
<couplelink islink="true" id="couple_52" from="person_5" to="couple_2" />
<person id="person_6">
  <property name="sex">male</property>
  <property name="sdm:y">166.39453</property>
  <property name="firstname">Henry</property>
  <property name="profession">Driver</property>
  <property name="sdm:x">391.375</property>
  <property name="name">Bilbot</property>
  <property name="birthdate">1945</property>
</person>
<childlink islink="true" id="child_2_6" from="couple_2" to="person_6" />
<person id="person_7">
  <property name="sex">male</property>
  <property name="sdm:y">26.648438</property>
  <property name="firstname">Jack</property>
  <property name="profession">Technician</property>
  <property name="deathdate">1997</property>
  <property name="sdm:x">214.79688</property>
  <property name="name">Johnson</property>
  <property name="birthdate">1925</property>
</person>
<couple id="couple_3">

```



```
<property name="startdate">1960</property>
<property name="sdm:y">73.23047</property>
<property name="sdm:x">305.84375</property>
<property name="status">married</property>
</couple>
<couplelink islink="true" id="couple_43" from="person_4" to="couple_3" />
<couplelink islink="true" id="couple_73" from="person_7" to="couple_3" />
</diagram>
```

In this XML file:

- ◆ A link is a top-level XML element (`childlink` or `couplelink`) with the attribute `isLink` set to `true` and attributes `from` and `to` which refer to nodes defined in the same file.
- ◆ All other top-level XML elements are treated as nodes (`person` and `couple`).
- ◆ Properties are defined as children of a node or link element. Each property has a value.
- ◆ The `sdm:x` and `sdm:y` properties are x- and y-coordinates giving the location of a node.

---

## Diagram format in XML

The root element of the diagram format in XML has the tag `diagram`.

In diagram format, there is an implicit data model. When data is read from an XML file, each XML element that is a child of the root element (each top-level element) is considered to be an object of the data model, specifically, a user-defined type.

The top-level elements are either links (attribute `isLink` is `true`) or nodes (attribute `isLink` is `false`). Nodes and links must be distinguishable by different XML element names. They are loaded from the same file.

The top-level XML elements that are not links (`isLink` attribute = `false`) are represented by nodes in the diagram. The elements that are links (`isLink` attribute = `true`) are represented by links in the diagram.

The name of the XML element (the string after the opening “<”) becomes the value of the tag property of the the model object.

The following attributes can be defined on an XML element:

- ◆ The XML attribute `id` represents the object’s unique identifier (ID).
- ◆ The XML attribute `isLink`, when set to `true`, indicates that the model object is a link.
- ◆ The XML attributes `from` and `to` for a link designate the origin and destination. The values of these attributes are the IDs of the origin and destination nodes.

The data properties of the objects appear in the XML file as child elements of the top-level elements with the name (tag) `property`.

The name of the property is the value of the `name` attribute. The value of the property is the text of the element.

For example, the following line shows a property `speed` with the value `148.0`:

```
<property name="speed">148.0</property>
```

---

## Metadata in the XML data file

You can include metadata in the XML data file to store information about the data in general and about specific objects.

The root `diagram` element has a Boolean attribute `metadata` which indicates whether metadata is to be handled. The default value is `true`.

```
<diagram metadata="true"/>
```

Metadata is represented by elements with the `sdm:` prefix, indicating a specific SDM namespace.

There are two types of metadata:

- ◆ Object-related
- ◆ General

### Object-related metadata

Object-related metadata can be included with a similar format to properties. The following example specifies a value for the metadata item `x`:

```
<property name="sdm:x">23</property>
```

### General metadata

General metadata can be included in a dedicated RDF section.

#### A general metadata section in an XML data file

```
<diagram metadata="true" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://dublincore.org/"
    xmlns:sdm="http://www.ilog.com/JViews/SDM/" />
<rdf:RDF>
  <rdf:Description about=".">
    <dc:Creator>
      <sdm:Pseudo>PhK</sdm:Pseudo>
      <sdm:Office>S023</sdm:Office>
    </dc:Creator>
  </rdf:Description>
</rdf:RDF>

... (the remainder of the document represents the SDM model)

</diagram>
```

Note that if an XML file contains an element with the name `rdf`, then the `metadata` attribute of the `diagram` element is forced to `true`.

---

## Application data in flat files for JViews Diagrammer

You can import application data from a flat file into the Designer.

The following examples, *Node data on Arizona cities in a flat file* and *Link data on traffic between Arizona cities in a flat file*, show the initial Arizona data files for nodes and links respectively as loaded in *Importing data from a flat file*. (The data is shown with spacing for readability.)

### Node data on Arizona cities in a flat file

```
City Availability Throughput latitude longitude Population
Flagstaff Online Busy 35D11'53N 111D39'02W 340000
Page Offline Unknown 36D84'00N 111D15'00W 30000
Tuba City Online Average 36D08'06N 111D14'21W 47600
Winslow Online Slow 35D01'27N 110D41'48W 90000
Sedona Online Slow 34D52'11N 111D45'37W 120000
Phoenix Online Busy 33D26'54N 112D04'24W 1200000
Tucson Online Busy 32D13'18N 110D55'33W 980000
Nogales Online Average 31D20'25N 110D56'01W 45000
Holbrook Online Average 34D54'08N 110D09'27W 16000
Show_Low Offline Busy 34D15'15N 110D01'45W 4000
Lake_Havasu Online Busy 34D29'02N 114D19'18W 20000
Taylor Online Slow 34D27'00N 110D06'00W 20000
Yuma Unknown Average 32D43'31N 114D37'25W 15000
Kingman Online Average 35D11'22N 114D03'08W 230000
```

### Link data on traffic between Arizona cities in a flat file

```
City1 City2 Traffic
Kingman Phoenix 2.6
Lake Havasu Yuma 1
Kingman Lake Havasu 2
Kingman Flagstaff 3.4
Tuba City Page 0.9
Flagstaff Tuba City 2.1
Flagstaff Winslow 2.7
Flagstaff Taylor 2.9
Flagstaff Sedona 1.1
Flagstaff Phoenix 4.3
Flagstaff Tucson 4.5
Holbrook Taylor 1.9
Show Low Taylor 0.9
Taylor Phoenix 3.1
Phoenix Tucson 4.5
Tucson Nogales 0.1
Yuma Phoenix 1.1
Sedona Flagstaff 2.1
Phoenix Phoenix 4
Nogales Phoenix 3.7
Flagstaff Tucson 5
```

A flat file is a text file containing the same number of values on each line. The values are separated by a separator character. The separator characters supported by the Designer are comma, semicolon, space, and tab. You can request autodetection of the separators. If a flat file contains values exported from Microsoft® Excel®, it is sometimes referred to as a CSV (comma-separated values) file.

In a flat file, the first line must contain the names of the data properties and each subsequent line must contain the property values for an object, with the properties in the same order as on the first line.

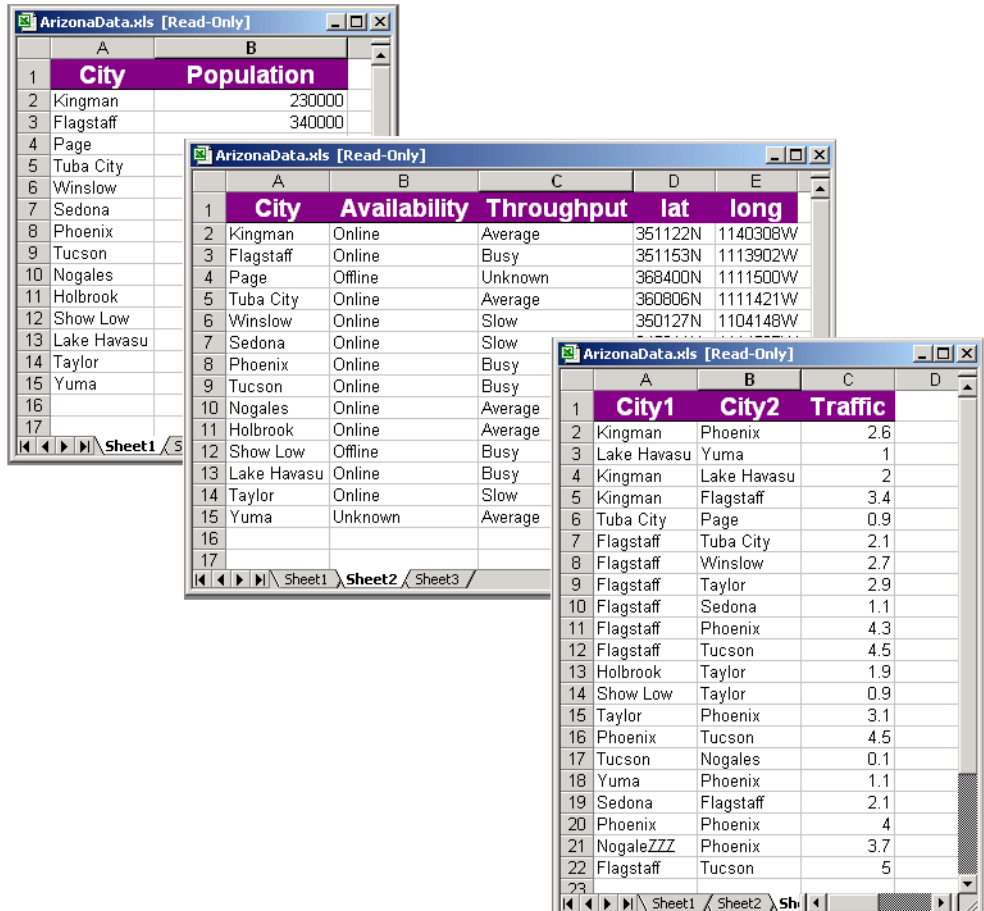
Nodes and links must be loaded from separate files.

You must define a mapping between the data properties in the file and the properties proposed by the Designer in the New Diagram Wizard. Some data properties may be unmapped and therefore not used in the diagram.

## Application data in databases for JViews Diagrammer

You can import application data from a database into JViews Diagrammer provided that the database is compliant with the JDBC API. The database must provide a JDBC driver and be accessible through SQL queries. For example, a Microsoft® Excel® file (.xls) is recognized as a database; the worksheets are treated as tables. A Microsoft® Access® database can also be used; the driver for both Microsoft® Excel® and Microsoft® Access® is called the JDBC-ODBC bridge.

The following figure shows the initial Arizona worksheets in Microsoft® Excel® as loaded in *Importing data from a database*.



Arizona data worksheets in Microsoft® Excel®

A database contains tables of columns and rows. Each row in a table represents an object in the data model. Each column in a table represents an attribute.

Nodes and links must be loaded from separate tables/sheets.

You must define a mapping between the columns in the file and the properties proposed by the Designer in the New Diagram Wizard. Some columns may be unmapped and therefore not used in the diagram.

---

## Project files

JViews Diagrammer project files (.idpr) are XML files. These small files contain a pointer to the style sheet file and a data source specification including a pointer to the XML file or the database connection configuration for a JDBC-compliant database.

The following example shows a project file when the data source type is XML.

### Project file: XML data source

```
<?xml version="1.0" encoding="UTF-8"?>
<diagrammer style="genealogy-composite.css">
  <datasource class="ilog.views.diagrammer.datasource.IlvXMLDataSource"
url="genealogy-composite.xml"/>
</diagrammer>
```

The following example shows a project file when the data source type is JDBC.

### Project file: JDBC data source

```
<?xml version="1.0" encoding="UTF-8"?>
<diagrammer style="ArizonaLogical.css">
  <datasource class="ilog.views.diagrammer.datasource.IlvJDBCDataSource">
    <connection driver="sun.jdbc.odbc.JdbcOdbcDriver" id="ID208245264"
passwd=""
url="jdbc:odbc:Driver={Microsoft Excel Driver (*.xls)};DBQ=C:\Ilog\
Repackaging\Builders\jviews-appl-sdm-builder\data\examples\ArizonaData.xls"
user=""/>
    <query connection="ID208245264" links="false" tag="node" value="select T1.
*, T2.Availability, T2.Throughput from [Sheet1$] as T1, [Sheet2$] as
T2 where T1.City = T2.City
">
      <map columns="0" property="id"/>
    </query>
    <query connection="ID208245264" links="true" tag="link" value="select *
from [Sheet3$]">
      <map columns="1" property="to"/>
      <map columns="0" property="from"/>
    </query>
  </datasource>
</diagrammer>
```



---

## Style sheet files

JViews Diagrammer style sheet files (.css) have the CSS for Java syntax. When you use the Designer to create a diagram component, you do not need to know CSS; when you save, the Designer creates the project file in XML and the style sheet file in CSS for Java™ .

The following code sample shows the style sheet that generates the diagram when you choose **File>New**:

### Example of style sheet for a new diagram

```
SDM {
    Map : "false" ;
    LinkLayout : "true" ;
    Composite : "true" ;
    GraphLayout : "true" ;
    HalfZooming : "false" ;
}
LinkLayout {
    _rule_name : "Link Layout" ;
    enabled : "true" ;
}

node {
    class : "@|symbolResource(ilog/views/palettes/shared/symbols/Rectangular.
css,Symbol)" ;
    Interactor : "ilog.views.sdm.IlvSDMCompositeInteractor" ;
    name : "@name" ;
}
link {
    class : "ilog.views.sdm.graphic.IlvGeneralLink" ;
    oriented : "true" ;
    foreground : "black" ;
    lineWidth : "1" ;
    arrowRatio : "3" ;
    maximumLineWidth : "2" ;
    minimumLineWidth : "1" ;
    arrowColor : "black" ;
}

link:selected {
    _rule_name : "is selected" ;
    foreground : "red" ;
    arrowColor : "red" ;
}
```

If you want to extend a diagram component by editing the style sheet manually, see Using CSS syntax in the style sheet in *Developing with the JViews Diagrammer SDK*.

---

## Next steps after the Designer

This topic shows you how to adapt your component and data configuration by using the Java™ API.

---

### Integrating the project file into an application

A project file created with the Designer includes styling and data source information. See *Integrating your development into an application* for more about project files and CSS style sheets.

The project file is used by an application through the `IlvDiagrammer` and `IlvDiagrammerProject` classes.

#### Typical Java code for including the project file in an application

```
IlvDiagrammer diagrammer = new IlvDiagrammer()
diagrammer.setProject (new IlvDiagrammerProject (new
    URL ("file:myproject.idpr"), true);
```

---

### Specifying a different data source

Once the project is loaded into the application, you can specify a data source other than the one contained by default in the project. This is an important feature, since the Designer is used with sample data that contains only enough information to specify the drawing style.

The following code example shows how to include a different data source.

#### Specifying a different data source

```
//Retrieve the project
IlvDiagrammer project = diagrammer.getProject();

//Create a new data source based on XML format
IlvXMLDataSource dataSource = new IlvXMLDataSource();
dataSource.setDataURL(new URL ("file:example.xml"));

//Associate new data source with current project
project.setDataSource (datasource);

//Update the diagrammer object with the newly created
//project
diagrammer.setProject (project);
```

---

### Customizing your application

At this stage, depending on your specific needs, there are several options you can consider for the architecture of your application:

- ◆ Replace the default data source with your own classes to feed the data model.

This option is easy to implement and provides the flexibility you need to accommodate your own type of data. The library contains default implementations for common data

sources, such as XML or JDBC. For other requirements, you can create your own implementation.

- ◆ Create your own data model and apply the specified style.

This option is recommended for advanced users who need to reuse existing objects and minimize overhead. The library contains several implementations of commonly used data models. For advanced requirements, you can write your own implementation of the `IlvSDMModel` interface.

Developing with the JViews Diagrammer SDK presents these different concepts in detail and guides you through different options and steps.

---

## Integrating the style sheets into an application

The Designer makes it much easier to write the style sheet. You can use the Designer for styling a sample of dummy data or an XML dump of your data. Then you can integrate the style sheet from the Designer into your application. You apply the styling defined in the style sheet to the real data used in the application.

To integrate an existing style sheet into an application:

1. Make sure that the project style sheet is accessible from the application.

The `.css` file must be accessible from the application.

2. In the application, after creating the diagram and loading the data model, use the following code:

```
try {
    diagram.setStyleSheets(new String[]{"..."});
} catch (IlvStylingException x) {
    System.err.println("Cannot load style sheets: " + x.getMessage());
}
```

See The style sheets in *Developing with the JViews Diagrammer SDK* for more information on how to load style sheets.



# Index

## Symbols

.css file for style sheet **61**

## A

Add button (for legend) **53**  
Add Column button **160**  
adding an object to the selection **170**  
adding new types of link **104**  
adding new types of node **103**  
adding objects **102**  
advanced graphic objects **124**  
application  
    launching **206**  
arguments  
    command-line **206**  
Arizona example  
    image files **152**  
arrow color **137**  
arrow heads **137**  
Arrow tab for link **137**

## B

background map **62**  
Basic tab for graph layout **130**

## C

cascading style sheets **24**  
changing  
    declarations **66**  
changing a style rule **45**  
Check Rules **119**  
CLASSPATH environment variable **206**  
closing a diagram **135, 136**  
collapsed **172**  
color settings **66**  
columns **182**  
Columns tab (for legend) **52**  
command-line arguments **206**  
copying style rules **115**

Create Style Rule Wizard **29, 41, 48**  
CSS

    debugging **69**  
    for a rule **116**  
    rule **68**

CSS code **116**

CSV (comma-separated values) **15**

CSV (comma-separated values) file **220**

Ctrl select **132**

custom data model **124**

customized legend **53**

customizer for styling **64**

customizing

    graphic objects **66**

cutting style rules **115**

## D

data

    in Excel **179**

    in memory **160**

data model

    custom **124**

    in a database **222**

    value **34**

Data Properties **172**

data properties

    displaying **165**

Data Properties toolbar **104**

Data Properties window **99**

data rules **109**

data source types **15**

database

    tables **222**

Database Mapping page **182**

database option **178**

Database page **179**

debugging

    style rules **120**

- style sheet **69**
- declaration
  - changing CSS **66**
- default legend **52**
- default link styling **25**
- default map **176**
- default node styling **24**
- default property values **24**
- default values **137**
- default values for properties **70**
- Delete Property button **105**
- deleting style rules **115**
- deployment **202**
- diagram
  - closing **135, 136**
  - printing **121, 212**
- diagram component
  - integrating **54**
- Diagram Editing button **102**
- Diagram Editing Mode **102**
  - grouping nodes **170, 171**
- diagram format (XML) **15**
- disabling style rules **117**
- Display tab (for legend) **52**
- dragging the mouse **170**

## E

- Edit button (for legend) **53**
- enabling style rules **117**
- enabling/disabling style rules **116**
- Enter an Expression **194**
- entering data values **160**
- Excel file **15, 179**
- expanded **172**
- Expression Assist **195**
- expressions **193**

## F

- filtering messages **120**
- filtering through a legend **53**
- flat file **15, 140**
- flat file option **140**
- Flow Direction parameter **130**
- footers **121**
- functions **197**

## G

- general links **81**
- general nodes **124**
- general rules **24**
- geometric form of links **131**
- Global link style **131, 132**
- graph layout **62**
  - Basic tab **130**
  - flow direction **130**
  - hierarchical **134**
  - Link Style tab **131**

- per-link properties **71**
- per-node properties **71**
  - selection **71**
- Graph Layout algorithms
  - parameters **128**
- Graph Layout option **132**
- graph layout theme **21**
- Graph Layout Wizard **134**
- Group Selection icon **170, 171**
- groups **170**

## H

- handle for expanding subgraph **62**
- headers **121**
- hierarchical algorithm **134**
- hierarchical layout **24**

## I

- icons
  - next to style rules **24, 25, 45**
- IlvDiagrammer class **226**
- IlvDiagrammerProject class **226**
- IlvSDMMModel interface **226**
- inheritance **68**
- initial look of diagram **19**
- integrating the diagram component **54**
- interactors
  - predefined **85**

## J

- JAR files **206**
- JDBC database **178**
- JViews Map Builder **176**
- JViews Maps **176**
- JViews Maps API **176**

## L

- latitude **62, 176**
- launching an application **206**
- layout **62, 71**
  - individual style **132**
- Layout Diagram **19**
- layout parameters **132**
- legacy support **124**
- legend **52, 62**
  - Add column **53**
  - Columns tab **52**
  - customized **53**
  - default **52**
  - Display tab **52**
  - Edit column **53**
  - Position tab **52**
- level of detail (high, medium, low) **65**
- link layout **62**
- Link Style parameter **132**
- Link Style tab for graph layout **131**
- link styles **132**

link styling  
    default **25**  
links  
    geometric form **131**  
    orthogonal **132**  
    style **132**  
Links tab **110**  
longitude **62, 176**

## M

map **62**  
    default **176**  
    positioning **176**  
    zooming **62**  
Map capability **176**  
Map Diagram **19**  
map file **176**  
mapping of database columns **182**  
messages  
    filtering **120**  
Messages panel **120**  
minus button **184**  
mixed link styles **132**  
Mixed links **132**  
mode  
    Diagram Editing **101**  
    Preview **101**  
    Style Editing **99**  
model value **34**  
modes **11**  
most specific style rule **45**  
moving objects **104**

## N

New Diagram Wizard **108**  
    on startup **11**  
New Property button **105**  
node  
    expanding **62**  
node styling  
    default **24**  
Nodes tab **110**

## O

Options **62**  
    Background Map **176**  
    Graph Layout **132**  
    Subgraphs **170**  
order of style rules **116**  
ordering style rules **116**  
Organization Chart Diagram **134**  
Orthogonal links **132**  
overlapping **62**  
overriding **68**

## P

Page Layout tab **121**

page numbers for printing **122**  
pages for printing **121**  
pages in New Diagram Wizard **108**  
panes  
    Style Rules **64**  
parameters for individual nodes or links **132**  
pasting a style rule **115**  
pasting style rules **115**  
PATH (UNIX) **11**  
per-link parameters **132**  
per-node parameters **132**  
plus button **184**  
Polyline links **131**  
Position tab (for legend) **52**  
positioning on a map **176**  
Preview button **51, 101**  
Preview Mode **51, 101**  
printing  
    footers **121**  
    headers **121**  
    page numbers **122**  
    pages **121**  
    zoomed print **121**  
printing facilities **121, 212**  
priority of rules **68**  
priority of style rules **116**  
project file **23**  
project file as output of Designer **124**  
properties  
    inherited values **67**  
    locally set **67**  
    per-node or per-link **71**  
properties of the data **165**  
property  
    in model **105**  
Property Sheet **67**  
property value  
    changing **99**  
property values  
    default **24**

## R

references to the data model **193**  
renaming style rules **115**  
resetting default property values **70**  
rule  
    priority **68**  
Rule Menu  
    Rename **115**  
    View **116**  
rule on all links **24**  
rule on all nodes **24**  
rule-select **99**  
rules  
    copying **115**  
    cutting **115**

- deleting **115**
- disabling **116**
- enabling **116**
- general **24**
- ordering **116**
- pasting **115**
- priority **68**
- renaming **115**
- viewing **115**
- rules on data **109**

**S**

- saving the diagram **23**
- Scaling **121**
- security items **179**
- selecting a rule **24, 99**
- selecting a style rule **24**
- selection
  - adding an object **170**
- selection box **45, 99**
  - thick **45**
  - thin **45**
- selector **68**
- separators in flat files **220**
- Set User-Defined Type button **104**
- Sheet tab **121**
- Shift-click **170**
- SQL query **179**
- Start menu (Windows®) **11**
- state
  - expanded/collapsed **172**
  - temporary **70**
- state of an object **137**
- Style Editing button **99**
- Style Editing Mode **99**
- style rule
  - changing **45**
  - most specific **45**
- style rules **24**
  - copying **115**
  - cutting **115**
  - debugging **120**
  - deleting **115**
  - enabling/disabling **116**
  - ordering **116**
  - pasting **115**
  - priority **68**
  - renaming **115**
  - selection **99**
  - testing **117**
  - viewing **115**
- style rules on data **109**
- Style Rules pane **64**
- style sheet **116**
  - debugging **69**
  - possible problems **69**

- Styling Customizer **64**
  - expressions **193**
  - references to model **193**
- Styling Customizers **66**
- styling properties **24**
- subgraphs **62, 170**
- Subgraphs option **170**
- symbol **24**

## T

- temporary change due to state **137**
- temporary state **70**
- testing style rules **117**
- thick selection box **45**
- thin selection box **45**
- toolbar
  - Data Properties **104**
- tooltips **51**

## U

- Undo command **118**
- undoing an action **118**
- UNIX **11**
- URL for database **179**
- user-defined option **158**

## V

- variables **196**
- viewing CSS **116**
- viewing style rules **115**

## W

- wizard
  - Create Style Rule **29, 41**
  - Graph Layout **134**
  - New Diagram **108**
- worksheets in Excel **222**

## X

- X/Y Diagram **19**

## Y

- yellow icon **24, 45**

## Z

- zoom thresholds **62**