IBM ILOG JViews Gantt V8.6

Using the Designer

Copyright Notices

Copyright notice

Trademarks

Notices

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

Table of contents

Getting Started

Explains how to create and style an activity-oriented or a resource-oriented Gantt chart.

In this section

Overview

This Getting Started shows you how to create and style an activity-oriented or a resource-oriented Gantt chart. The data used is in XML that conforms to the DTD used in the Schedule Data eXchange Language (SDXL).

Running the Designer

You can run the Designer from Microsoft® Windows® or UNIX® systems.

To run the Designer on a Microsoft Windows system:

Choose, for example, **Start > Programs > IBM ILOG > IBM ILOG JViews Gantt 86 > Designer** where IBM ILOG is the default program group suggested by the installer,

or

Double-click `run.bat` in the directory:

```
<installdir>\jviews-gantt86\bin\designer
```

To run the Designer on a UNIX® system, run the shell script `run.sh` in the directory:

```
<installdir>/jviews-gantt86/bin/designer
```

Developing an activity-oriented Gantt chart

Explains what an activity-oriented Gantt chart is and how to use the New Gantt Chart Wizard.

In this section

Overview

An activity-oriented Gantt chart shows activities and subactivities along the y axis against time on the x axis. Each activity lasts for a projected amount of time, its duration. A dependency between activities, for example, where an activity cannot be started before another activity is completed, is shown by a constraint. A milestone in the schedule represented by this type of Gantt chart is represented by a special kind of activity that has no duration: zero-duration activity.

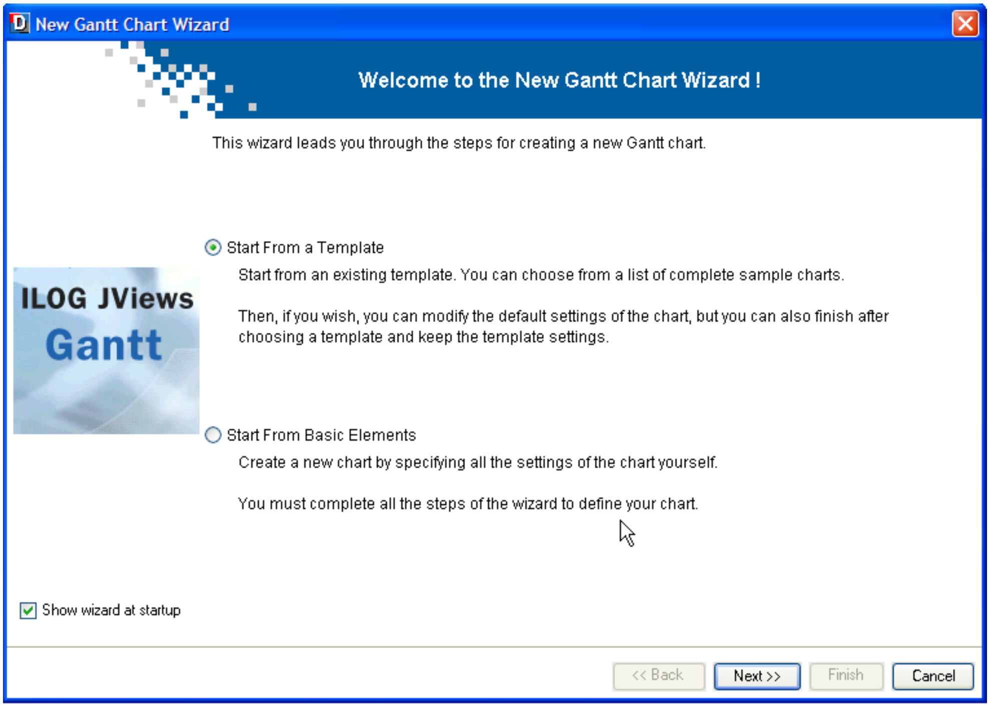
This type of Gantt chart is implemented by an instance of the class `IlvGanttChart` in the Java API. See The Gantt beans in *Developing with the SDK*.

Creating an activity-oriented Gantt chart

The easiest way to create a new Gantt chart is through the New Gantt Chart Wizard. The first time that you start the Designer, the New Gantt Chart Wizard is displayed.

If you do not want this wizard to be displayed automatically at startup, clear the option. You can access the New Gantt Chart Wizard from the menu bar (**File>New from Wizard**) or by typing **Ctrl-W**.

The following figure shows the New Gantt Chart Wizard.



New Gantt Chart Wizard

Welcome to the New Gantt Chart Wizard !

This wizard leads you through the steps for creating a new Gantt chart.

● Start From a Template

Start from an existing template. You can choose from a list of complete sample charts.

Then, if you wish, you can modify the default settings of the chart, but you can also finish after choosing a template and keep the template settings.

○ Start From Basic Elements

Create a new chart by specifying all the settings of the chart yourself.

You must complete all the steps of the wizard to define your chart.

☑ Show wizard at startup

In this section you will:

- Create a new Gantt chart project with the basic elements of a Gantt chart.
- Choose an XML-type data source for importing your data into the chart.
- Choose the XML file that contains the data for the basic Gantt chart.
- Select the activity-oriented Gantt chart, a basic time-and-activity bar chart; there is no indication of the resources that will be applied to the tasks represented in this chart.
- Choose a theme to determine a ready made look and feel for the chart.

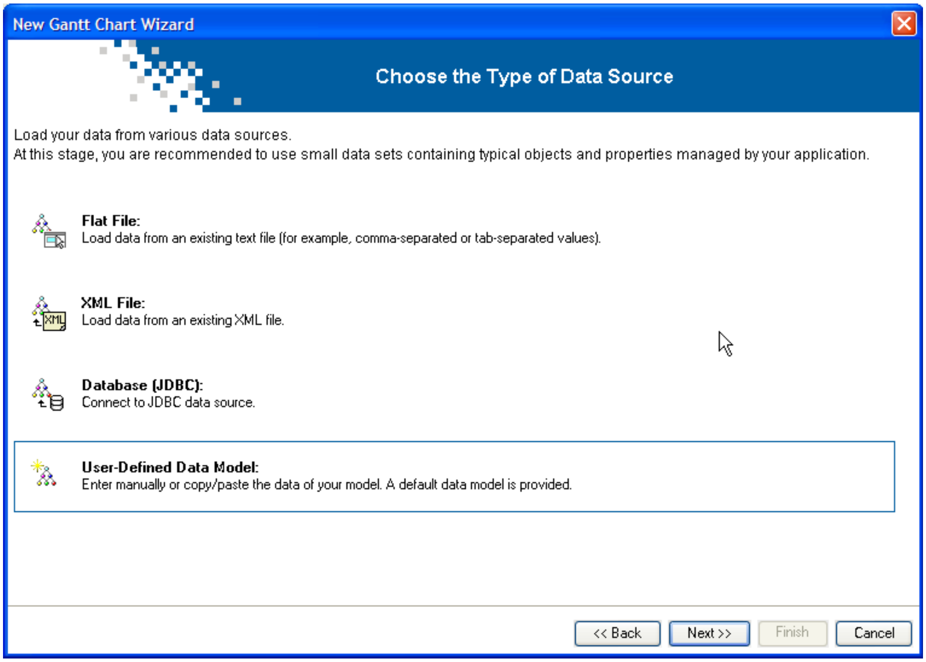
The objective is to set up your chart quickly and easily, so that you can start customizing the rendering of the chart and its data through the styling facilities of JViews Gantt.

The Gantt chart you create will show completion of the activities and whether the completion is in advance of or behind the scheduled completion dates of each activity.

Starting from the basic elements

In this example, you are going to create your own project file (extension `.igpr`) to contain the data and styling of your chart. The project file holds the basic settings of a chart, which are, the chart itself, the table , the sheet , the time scale, the horizontal grid and the vertical grid.

1. Select **Start From Basic Elements**.
2. Click **Next**.



Load your data from various data sources.
At this stage, you are recommended to use small data sets containing typical objects and properties managed by your application.

Flat File:
Load data from an existing text file (for example, comma-separated or tab-separated values).

XML File:
Load data from an existing XML file.

Database (JDBC):
Connect to JDBC data source.

User-Defined Data Model:
Enter manually or copy/paste the data of your model. A default data model is provided.

Choosing a type of data source

You are going to use existing XML data in the new Gantt chart.

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

Loading data from an Existing Gantt XML file

The data that the file contains is compliant with the Gantt data model. Important points to note for the particular features that this file allows you to implement are:

- The property `completion`, which allows you to show which activities are completed.

- The user-defined type (`"tags"`) named `critical`, which allows you to show which activities are on a critical path towards completion and may not be completed on time.

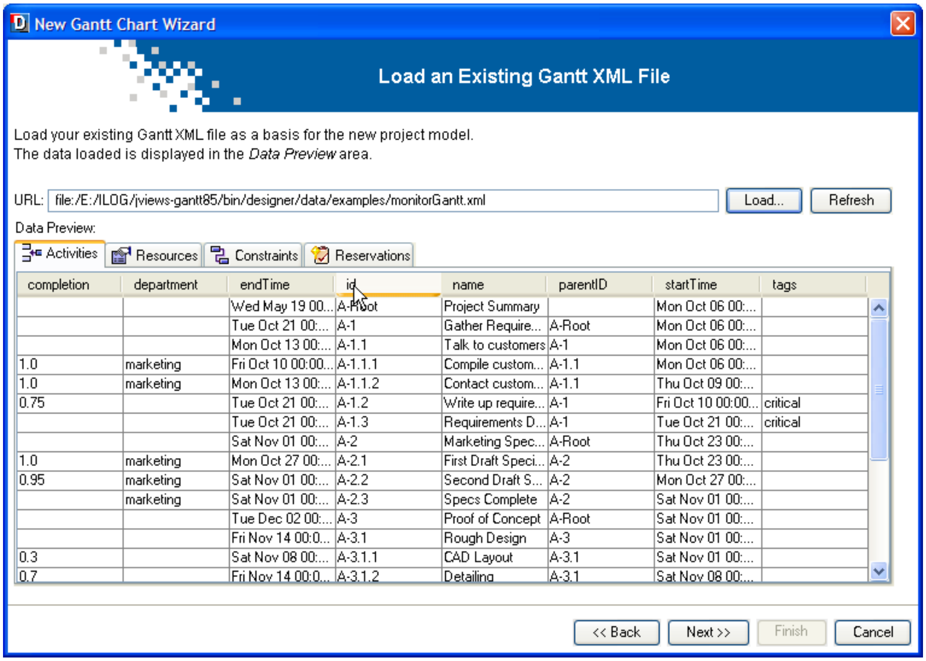
You can later customize the styling of completed and critical activities to give them a different appearance.

See *Content and structure of an XML data file* for an explanation of the data content and structure of `monitorGantt.xml`. In particular, see *Activities* and *Constraints*.

1. Click **Load** to browse for the XML file that contains the data for this example.
2. Select the file `monitorGantt.xml` from the `examples` directory and click **Open**.

   The XML file is validated and loaded. You have a preview of a read-only `JTable` view of the data (**Data Preview**), open by default on the **Activities** tab. This view shows the XML data corresponding to the `activity` objects in the data model.

   The following figure shows the Activities view of XML data.



Load your existing Gantt XML file as a basis for the new project model.
The data loaded is displayed in the *Data Preview* area.

URL: file:/E:/ILOG/jviews-gantt85/bin/designer/data/examples/monitorGantt.xml

Data Preview:

completion	department	endTime	id	name	parentID	startTime	tags
		Wed May 19 00...	A-Root	Project Summary		Mon Oct 06 00:...	
		Tue Oct 21 00:...	A-1	Gather Require...	A-Root	Mon Oct 06 00:...	
		Mon Oct 13 00:...	A-1.1	Talk to customers	A-1	Mon Oct 06 00:...	
1.0	marketing	Fri Oct 10 00:00...	A-1.1.1	Compile custom...	A-1.1	Mon Oct 06 00:...	
1.0	marketing	Mon Oct 13 00:...	A-1.1.2	Contact custom...	A-1.1	Thu Oct 09 00:...	
0.75		Tue Oct 21 00:...	A-1.2	Write up require...	A-1	Fri Oct 10 00:00...	critical
		Tue Oct 21 00:...	A-1.3	Requirements D...	A-1	Tue Oct 21 00:...	critical
		Sat Nov 01 00:...	A-2	Marketing Spec...	A-Root	Thu Oct 23 00:...	
1.0	marketing	Mon Oct 27 00:...	A-2.1	First Draft Speci...	A-2	Thu Oct 23 00:...	
0.95	marketing	Sat Nov 01 00:...	A-2.2	Second Draft S...	A-2	Mon Oct 27 00:...	
	marketing	Sat Nov 01 00:...	A-2.3	Specs Complete	A-2	Sat Nov 01 00:...	
		Tue Dec 02 00:...	A-3	Proof of Concept	A-Root	Sat Nov 01 00:...	
		Fri Nov 14 00:0...	A-3.1	Rough Design	A-3	Sat Nov 01 00:...	
0.3		Sat Nov 08 00:...	A-3.1.1	CAD Layout	A-3.1	Sat Nov 01 00:...	
0.7		Fri Nov 14 00:0...	A-3.1.2	Detailing	A-3.1	Sat Nov 08 00:...	

3. Click the **Resources**, **Constraints**, and **Reservations** tabs to view the data corresponding to these data model objects..
4. Click **Next**.

Previewing the mapping

Data from an XML data source must comply with the DTD used in the Schedule Data eXchange Language (SDXL). See *XML structure* for more information. The mapping of data in an SDXL-compliant file is done automatically. You can view the mapping in this step of the New Gantt Chart Wizard, but you cannot change it.

The mandatory properties of activities are shown in blue. They are mapped to the corresponding tables in the data source. These tables are shown as columns in the lower part of the **Activities Table** pane. (Activities are shown in the activity-oriented Gantt chart.)

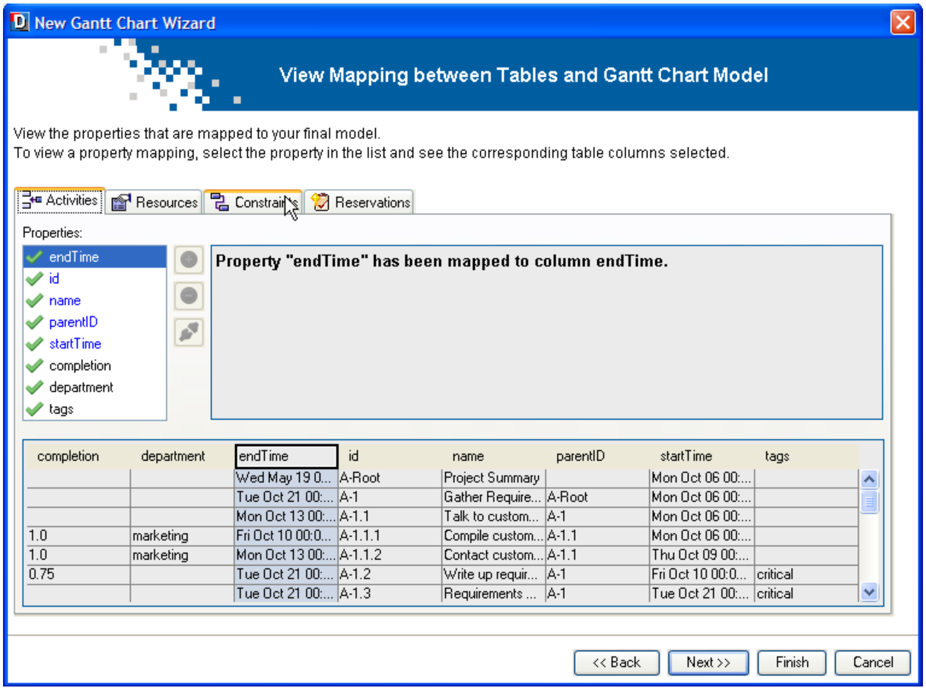
User-defined properties are shown in black. They are also mapped to the corresponding tables in the data source. User-defined properties are properties that are not defined in the default Gantt data model. See *User-defined properties* for how to define these properties in an XML file.

1. Click the **Resources Table** tab to see the mapping of the properties defined for resources in `monitorGantt.xml`. Resources are shown in the resource-oriented Gantt chart (or Schedule chart).

2. Click the **Constraints Table** tab to see the mapping of the properties defined for constraints in `monitorGantt.xml`. Constraints are shown in the activity-oriented Gantt chart.

3. Click the **Reservations Table** tab to see the mapping of the properties defined for reservations in `monitorGantt.xml`. Reservations are shown in the resource-oriented Gantt chart (or Schedule chart).

   The following figure shows data mapping for XML.



View Mapping between Tables and Gantt Chart Model

View the properties that are mapped to your final model.
To view a property mapping, select the property in the list and see the corresponding table columns selected.

Properties:

- endTime
- id
- name
- parentID
- startTime
- completion
- department
- tags

Property "endTime" has been mapped to column endTime.

completion	department	endTime	id	name	parentID	startTime	tags
		Wed May 19 0...	A-Root	Project Summary		Mon Oct 06 00:...	
		Tue Oct 21 00:...	A-1	Gather Require...	A-Root	Mon Oct 06 00:...	
		Mon Oct 13 00:...	A-1.1	Talk to custom...	A-1	Mon Oct 06 00:...	
1.0	marketing	Fri Oct 10 00:0...	A-1.1.1	Compile custom...	A-1.1	Mon Oct 06 00:...	
1.0	marketing	Mon Oct 13 00:...	A-1.1.2	Contact custom...	A-1.1	Thu Oct 09 00:...	
0.75		Tue Oct 21 00:...	A-1.2	Write up requir...	A-1	Fri Oct 10 00:0...	critical
		Tue Oct 21 00:...	A-1.3	Requirements ...	A-1	Tue Oct 21 00:...	critical

4. Click **Next**.

Selecting the type of chart

You need to select the appropriate type of chart for displaying the data. The emphasis is on the tasks to perform, their duration, milestones in their performance, and any constraints that might affect completing the tasks on time.

1. Select the activity-oriented Gantt chart.

   This Gantt chart displays the data from the XML file representing activities, milestones, and constraints.

2. Click **Next**.

Choosing a theme

You can select an appropriate theme for your project from the list of available themes.

Themes provide a particular look and feel by specifying, for example, the fonts, vertical and horizontal grids, main bar, and background to use. A theme is a style sheet that is available as a starter style sheet. It is a convenient way of applying a common look that you want to reuse in different projects.

The themes provided apply styles to the chart itself and to the mandatory properties of the data it contains. The mandatory properties for the start time, end time, ID, and name can be styled in a theme.

1. Select **A green theme**.

   This theme colors important subcomponents of the chart and elements of its data representation in different shades of green and uses large font sizes.

2. Click **Finish**.

Congratulations. You have completed the setup of your new chart. Click **Finish**.

You have a complete chart containing the data specified by the XML file. You can customize the styling of this chart and its data in the Designer.

The chart as it stands now should be the same as a chart derived from the **basicGantt** template with **A green theme** applied.

Customizing Gantt chart Options

In the Designer you make these modifications in **Style Editing** mode.

Changing the chart appearance

This section shows you how to change the row height and the move mode of the divider.

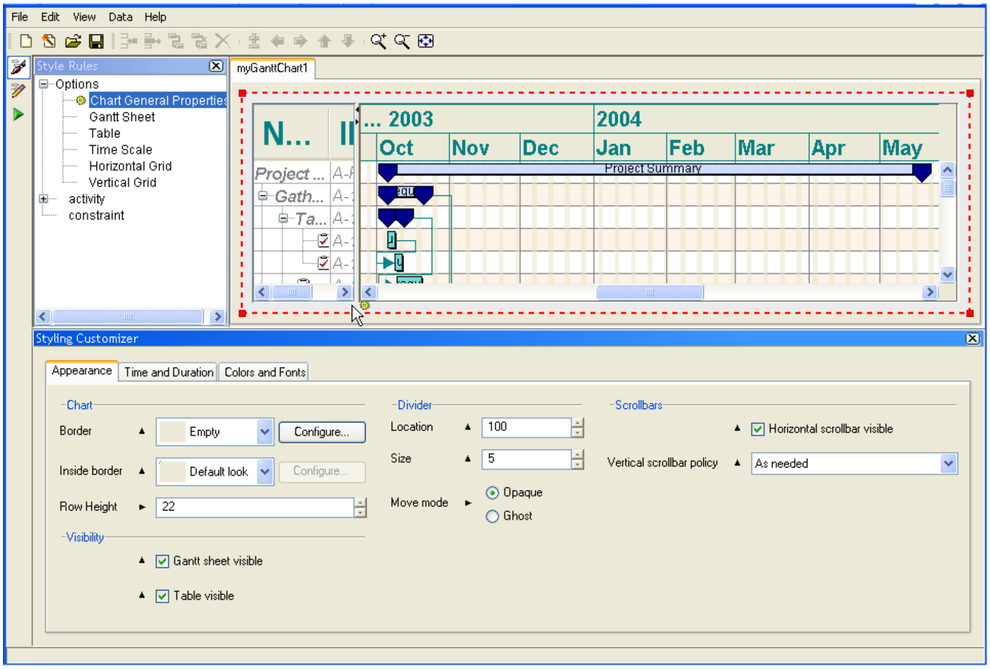
1. Expand Options under **Style Rules**.
2. Select **Chart General Properties** to modify the chart component.
3. Make sure the **Appearance** tab is active in the **Styling Customizer**.
4. Decrease the **Row Height** to **22**.

   The rows in the Gantt chart get narrower.

5. Select **Opaque** as the **Move mode**.

   The move mode controls the visual effect of moving the divider between the table and the Gantt Sheet. Making the move mode opaque means that the table and the Gantt Sheet are resized continuously as you move the mouse.

Your Gantt chart should look like the following figure, it shows the settings in the Styling Customizer.



Chart

Border: Empty

Inside border: Default look

Row Height: 22

Visibility

☑ Gantt sheet visible

☑ Table visible

Divider

Location: 100

Size: 5

Move mode: ● Opaque ○ Ghost

Scrollbars

☑ Horizontal scrollbar visible

Vertical scrollbar policy: As needed

Changing features of the Gantt table

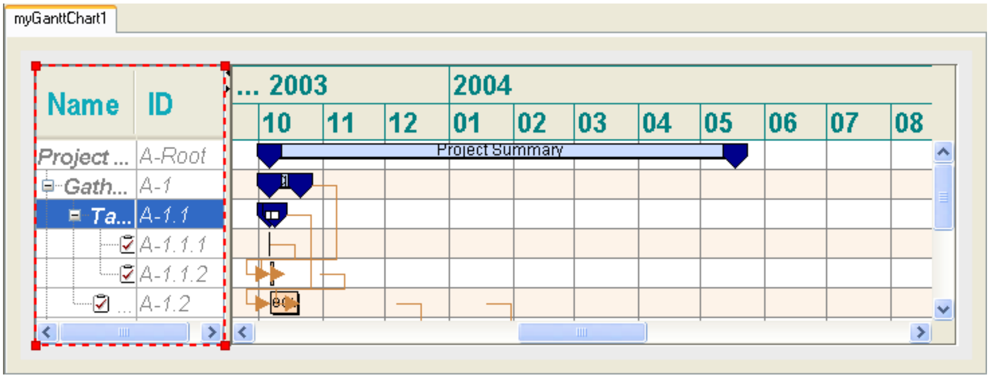
This section shows you how to change the font and the font size used in the table header and change the header background color.

1. Select the **Table** subcomponent.
2. Make sure the **Appearance** tab is active in the **Styling Customizer**.
3. In the **Table Header** section, click the ellipsis (**...**) in the **Font** field.
4. In the **Font Editor** window select the font **Arial Narrow** and the font size **24**.

   The effect of the changes is shown in the **Preview** area.
5. Click **Apply**.
6. Click the ellipsis (**...**) in the **Foreground color** field.
7. In the **Color Editor** window select the **RGBA** tab.
8. Change the **Red** setting to **14**, the **Green** setting to **170**, and the **Blue** setting to **186**.

   The effect of the new color setting is shown in the **Preview** area. You have set the header foreground to a shade of turquoise.
9. Click **Apply**.

The following figure shows how your Gantt chart should look at this stage. It displays the table with modified header font and foreground color.



Name	ID
Project ...	A-Root
Gath...	A-1
Ta...	A-1.1
☑	A-1.1.1
☑	A-1.1.2
☑ ...	A-1.2

Changing the time scale

This section shows you how to change the font and the font size used in the time scale and change the foreground color.

1. Select the **Time Scale** subcomponent.
2. Click the ellipsis (**...**) in the **Font** field.
3. In the **Font Editor** window select the font **Arial Narrow** and the font size **14**.

   The effect of the changes is shown in the **Preview** area.

4. Click **Apply**.

5. Click the ellipsis (**...**) in the **Foreground color** field.

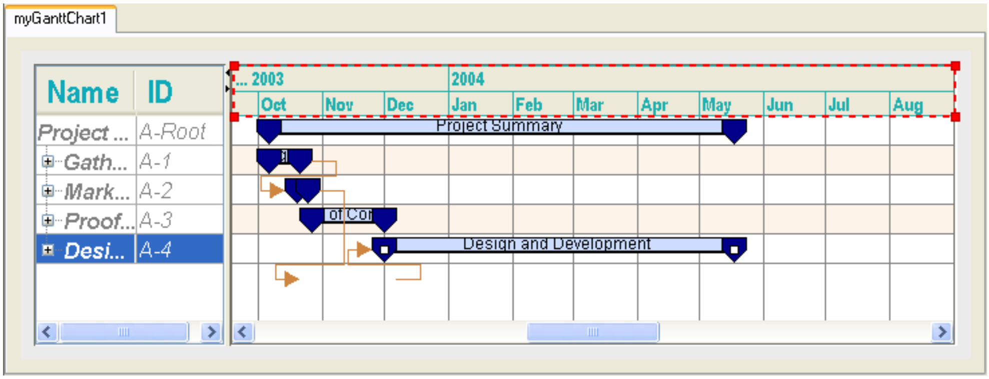
6. In the **Color Editor** window select the **HSB** tab.

7. Change the **H** setting to **179**, the **S** setting to **82**, and the **B** setting to **69**.

   The color defined by the new HSB settings is shown in the **Preview** area and should be similar to the turquoise color selected for the foreground of the table header with the RGB coordinates.

8. Click **Apply**.

The following figure shows how your Gantt chart should look at this stage. It shows the time scale with modified font and foreground color.



Name	ID
Project ...	A-Root
⊞ Gath...	A-1
⊞ Mark...	A-2
⊞ Proof...	A-3
⊞ Desi...	A-4

Changing the horizontal grid

1. Select the **Horizontal Grid** subcomponent.

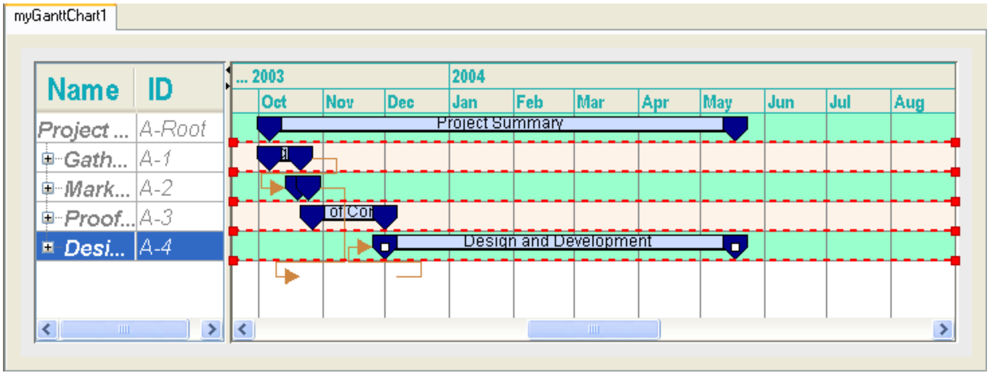
2. In the **Styling Customizer** click the ellipsis (**...**) in the **Odd Rows Color** field.

3. In the **Color Editor** window select the **Swatches** tab.

4. Select the pale peppermint swatch that corresponds to the setting (**153,255,204**).

5. Click **Apply**.

The following figure show what you Gantt chart should look at this stage. It displays a horizontal grid with modified fill color for odd rows.



Name	ID
Project ...	A-Root
Gath...	A-1
Mark...	A-2
Proof...	A-3
Desi...	A-4

Saving a project

It is important to save your work periodically to avoid losing it.

- To save the chart initially, choose **File > Save As** and enter a file name in the file chooser. When you save your work, the Designer saves three files: a project file, a style sheet, and a data file.

The project file specifies the name of the style sheet file, the type and URL of the data source, and the name of the data file.

The **File > Save As** and **File > Save** commands allow you to save the project file with the extension `.igpr`. The other two files are saved implicitly.

You can find examples of project files in the directory:

```
<installdir>/jviews-gantt86/bin/designer/data/examples
```

Styling the data

The chart you created in the New Chart Wizard had the styling of the basic Gantt chart. You can customize the styling of the data in the chart by adding to the style sheet. The style sheet is a `.css` file that contains style rules that define the styling of the data. You can create or change a style rule in the Create Style Rule Wizard or the Change Style Rule Wizard.

The currently defined style rules can apply to instances of activity or constraint. You can view them in the hierarchy listed under Style Rules. There are rules that control the styling of activities in general, parent activities lower in the hierarchy, leaf activities, and milestones. There is also a style rule for constraints. The names of these rules are:

- `activity`
- `activity:parent`
- `activity:leaf`
- `activity:milestone`
- `constraint`

Applying general and specific rules

There is one rule defined for all activities in the Gantt chart. This rule sets the default properties of all activities for the top and bottom margins, the tooltip, and the shape and label properties.

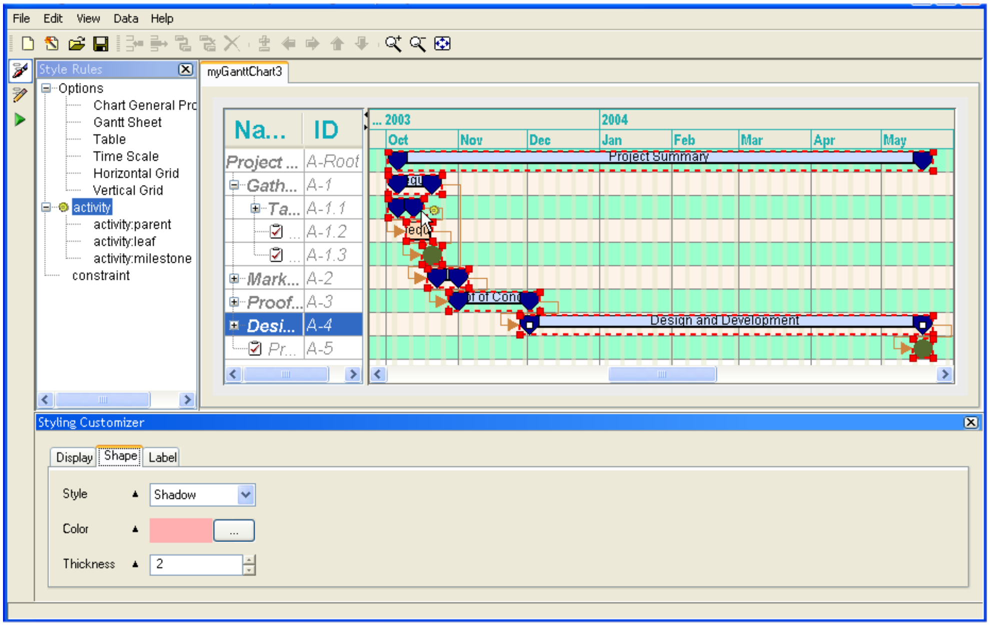
An activity is a task that must be completed. Activities are hierarchical in nature. The hierarchy consists of parent activities and child activities, which can themselves be parent activities, right down to leaf activities. A leaf activity is an activity that has no child activity.

When you select the `activity` entry in the Style Rules, all activities to which this general rule applies are selected in the Gantt chart. You can see the default properties set by this rule in the Styling Customizer.

There can be only one general rule at the top level of the hierarchy of style rules that control the styling of activities. If you try to create another `activity` rule, you are prompted to delete the existing one.

The yellow icon beside the style rule indicates that this rule applies to the currently selected objects. It is the only currently applicable rule, since none of the other style rules have a yellow icon beside them. It is a general rule that can be overridden by more specific rules that appear later and therefore lower in the hierarchy. This is why the parent activities in the Gantt chart look different and do not seem to display the same properties as defined by the selected activity rule.

The following figure shows the default properties of the style rule for all activities.



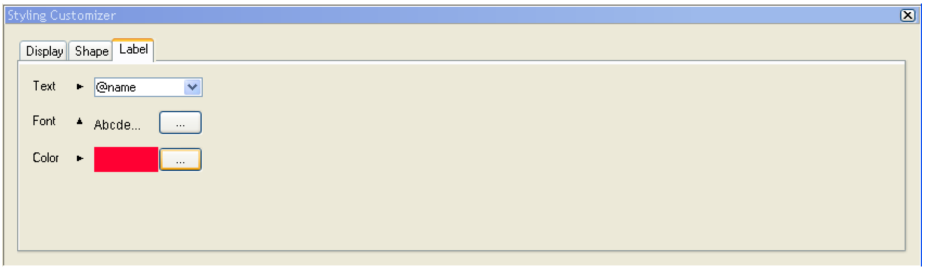
Style ▲ Shadow

Color ▲ ...

Thickness ▲ 2

- Select the rule `activity:leaf` and examine its properties in the Styling Customizer. It has the same properties as the `activity` rule, but the values are different and the leaf activities therefore look different from the expected result of the `activity` rule.

  The following figure shows the label properties set for the leaf activities by the style rule `activity:leaf`. Select the `activity` style rule and click the **Label** tab in the Styling Customizer. You can see that the default label properties of activities do not all include the `@name` property.



Text ► @name

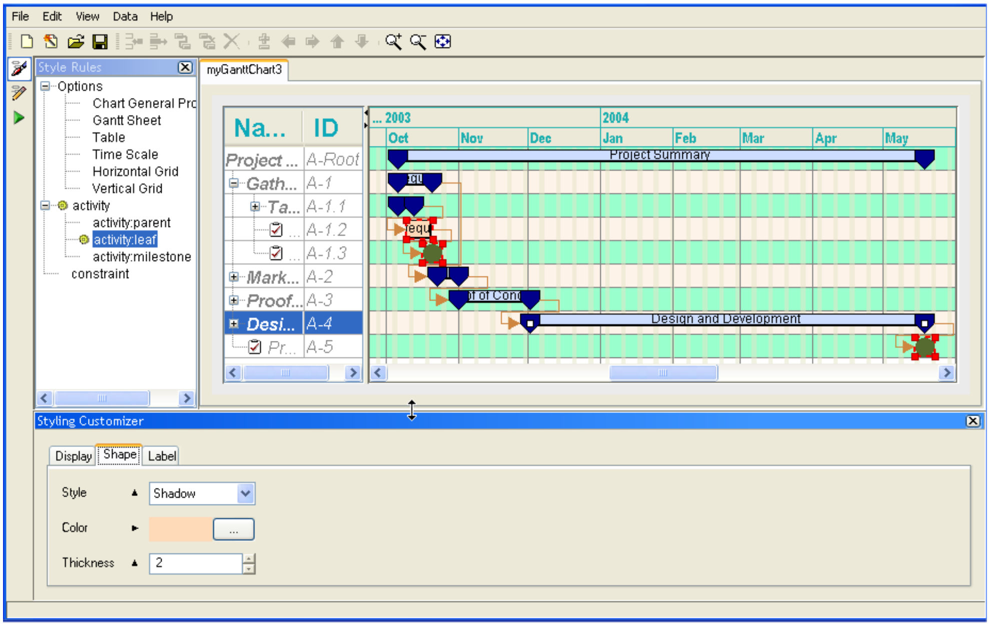
Font ▲ Abcde... ...

Color ► ...

When you select the rule `activity:leaf`, another yellow icon appears beside this rule. The yellow icons indicate that the rules `activity` and `activity:leaf` are currently applicable.

The last style rule created overrides other applicable rules. The more recent a style rule, the lower in the hierarchy it appears. You can move rules in the hierarchy. If you move a rule up in the hierarchy, it has a lower priority and can be overridden by style rules lower in the hierarchy in contexts in which all these rules apply.

The following figure shows that a different color is set for the shape of leaf activities. (You need to switch to Gantt Chart Editing Mode to expand the activities in the Gantt chart and view the leaf activities themselves.)

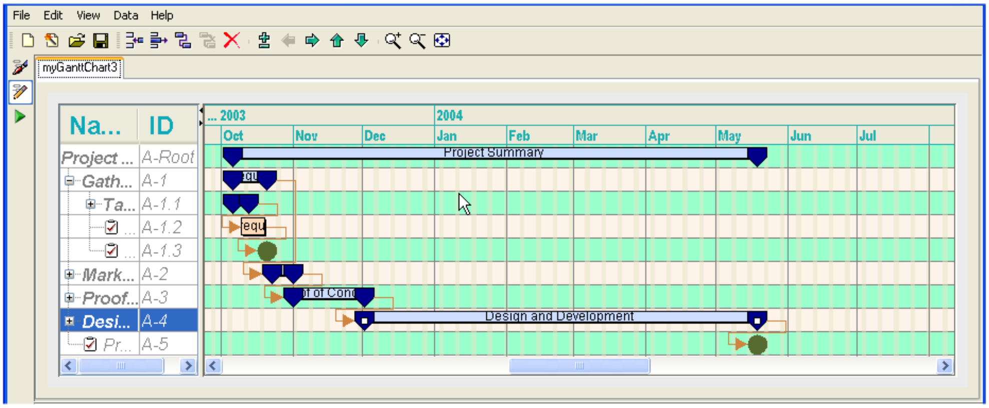


Style ▲ Shadow

Color ▸ ...

Thickness ▲ 2

The following figure shows the expanded leaf activities in Gantt chart editing mode.



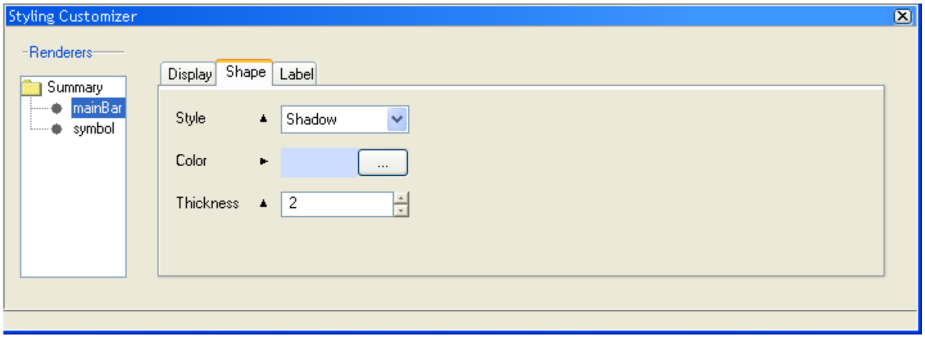
Styling with a summary renderer

Select the style rule `activity:parent`. This rule controls the styling of the parent activities in the Gantt chart. When you select the rule `activity:parent`, a yellow icon appears beside this rule. The yellow icons indicate that the rules `activity` and `activity:parent` are currently applicable to the selected objects in the Gantt chart.

A parent activity is represented in the Gantt chart by a collection of basic renderers, known as a summary renderer. The summary renderer contains an activity bar renderer and a symbol renderer used for the start and end times of the activity.

1. In the Styling Customizer, select the **mainBar** renderer in the summary tree. Click the **Shape** tab. *[no title]* shows the properties used to render the shape of the activity bar of parent activities.
2. Select the **symbol** renderer to see the properties used to render the shape of the start and end time symbols of parent activities.

   The following figure shows the renderer for styling activity bar of parent activities.

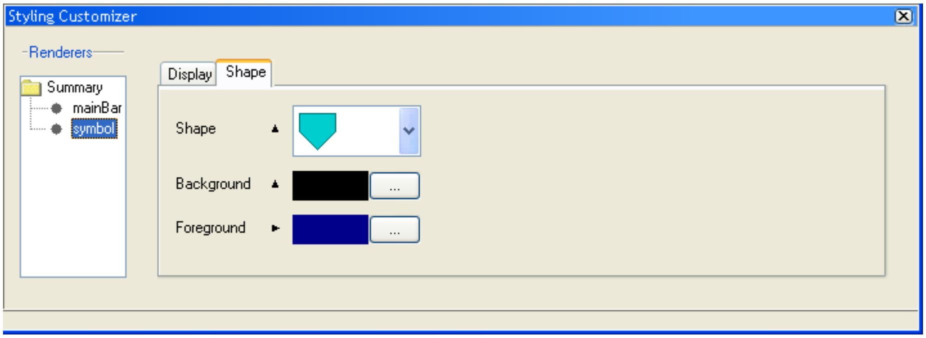


Style ▲ Shadow

Color ►

Thickness ▲ 2

   The following figure shows the renderer for styling start and end time symbols of parent activities.



Shape ▲

Background ▲

Foreground ►

See *Change style rule* for how to change the rendering applied by this style rule.

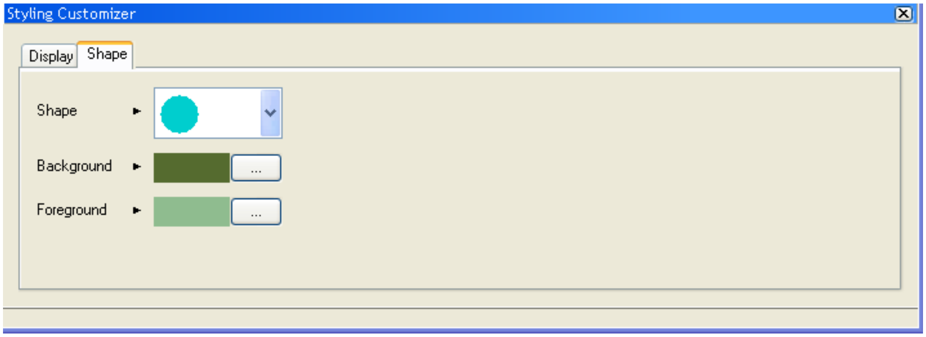
See *Create a style rule condition for specific parent activities* for how to write a more specific new style rule for controlling the representation of parent activities.

Styling milestones

- Select the rule `activity:milestone`. You can see in the Styling Customizer that this rule controls the Display properties and the Shape properties for representing milestones. A milestone is a zero-duration activity, that is, it has the same start and end time.

When you select the rule `activity:milestone`, a yellow icon appears beside this rule. The yellow icons indicate that the rules `activity` and `activity:milestone` are currently applicable to the selected object in the Gantt chart.

The following figure shows the Shape properties of milestones.


Styling Customizer

Shape ▸ ●

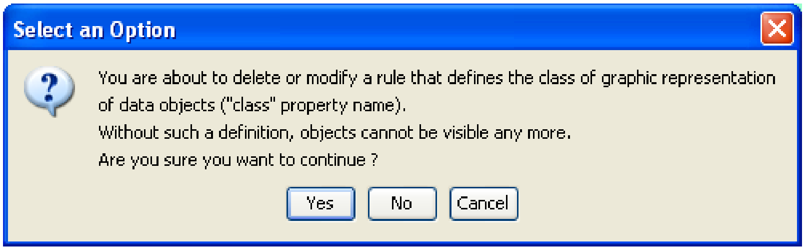
Background ▸ 🟩 ...

Foreground ▸ 🟩 ...

Change style rule

You can change the representation of activities in the Gantt chart by changing the renderer in the style rule `activity:parent`. You can do this in the Change Style Rule Wizard.

1. Select the style rule `activity:parent`.
2. Choose **Edit > Change Style Rule** or right-click and choose **Change Style Rule**.
3. Click **Yes** to confirm that you want to modify the style rule governing the graphic representation of the selected class of object.

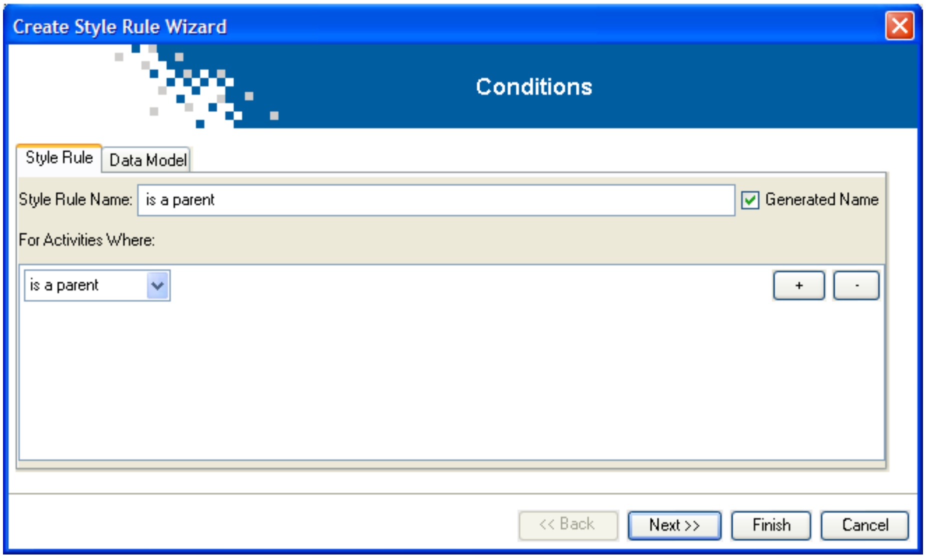

Select an Option

You are about to delete or modify a rule that defines the class of graphic representation of data objects ("class" property name).
Without such a definition, objects cannot be visible any more.
Are you sure you want to continue ?

4. Click **Next** on the Welcome page of the Change Style Rule Wizard.

   If you do not want this page to be displayed when you open the wizard, clear the option.

   The following figure shows how the condition (left part) of the style rule `activity:parent` is displayed. The condition determines the context in which the styling will be applied. You can read it as "If the object is of type activity and it has no user-defined type property and it is a parent activity and it has no specific attribute".



Style Rule Name: is a parent ☑ Generated Name

For Activities Where:

is a parent

A generated name is applied to the style rule that is to be modified. In this case, the name is "is a parent". This name will appear in the Style Rules tree when you have finished restyling the graphic representation applied by the style rule. If you want, you can change the name to one of your own choice.

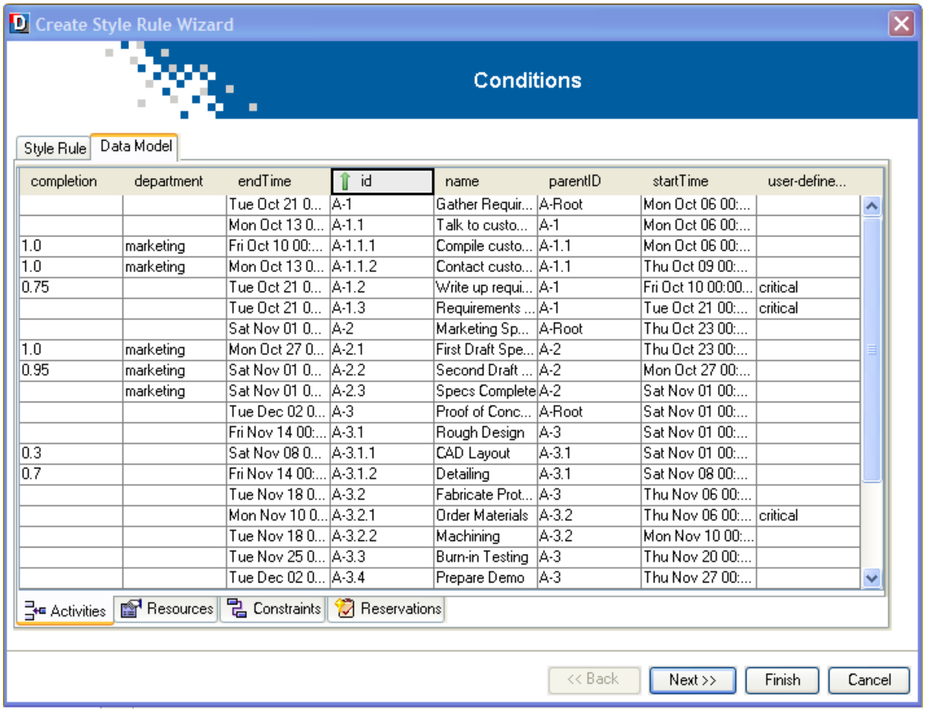
Since you will change only the representation of objects to which the rule applies and you will leave the condition unchanged, it is better to retain the original name.

5. Clear the option **Generated Name** and enter the original name of the style rule `activity:parent`.

6. Click the **Data Model** tab to have a read-only view of the data model.

   It is useful to remind yourself of the specific properties of objects when you want to change the condition of a style rule or write a new style rule. The following figure shows you the Activities table view of the Gantt Data Model. The columns represent the properties defined for `activity` objects. You can see that the only user-defined type is `critical`.

   This example is not concerned with changing the condition part of the style rule, but only with changing the styling applied through the rule.



completion	department	endTime	id	name	parentID	startTime	user-define...
		Tue Oct 21 0...	A-1	Gather Requir...	A-Root	Mon Oct 06 00:...	
		Mon Oct 13 0...	A-1.1	Talk to custo...	A-1	Mon Oct 06 00:...	
1.0	marketing	Fri Oct 10 00:...	A-1.1.1	Compile custo...	A-1.1	Mon Oct 06 00:...	
1.0	marketing	Mon Oct 13 0...	A-1.1.2	Contact custo...	A-1.1	Thu Oct 09 00:...	
0.75		Tue Oct 21 0...	A-1.2	Write up requi...	A-1	Fri Oct 10 00:00...	critical
		Tue Oct 21 0...	A-1.3	Requirements ...	A-1	Tue Oct 21 00:...	critical
		Sat Nov 01 0...	A-2	Marketing Sp...	A-Root	Thu Oct 23 00:...	
1.0	marketing	Mon Oct 27 0...	A-2.1	First Draft Spe...	A-2	Thu Oct 23 00:...	
0.95	marketing	Sat Nov 01 0...	A-2.2	Second Draft ...	A-2	Mon Oct 27 00:...	
	marketing	Sat Nov 01 0...	A-2.3	Specs Complete	A-2	Sat Nov 01 00:...	
		Tue Dec 02 0...	A-3	Proof of Conc...	A-Root	Sat Nov 01 00:...	
		Fri Nov 14 00:...	A-3.1	Rough Design	A-3	Sat Nov 01 00:...	
0.3		Sat Nov 08 0...	A-3.1.1	CAD Layout	A-3.1	Sat Nov 01 00:...	
0.7		Fri Nov 14 00:...	A-3.1.2	Detailing	A-3.1	Sat Nov 08 00:...	
		Tue Nov 18 0...	A-3.2	Fabricate Prot...	A-3	Thu Nov 06 00:...	
		Mon Nov 10 0...	A-3.2.1	Order Materials	A-3.2	Thu Nov 06 00:...	critical
		Tue Nov 18 0...	A-3.2.2	Machining	A-3.2	Mon Nov 10 00:...	
		Tue Nov 25 0...	A-3.3	Burn-in Testing	A-3	Thu Nov 20 00:...	
		Tue Dec 02 0...	A-3.4	Prepare Demo	A-3	Thu Nov 27 00:...	

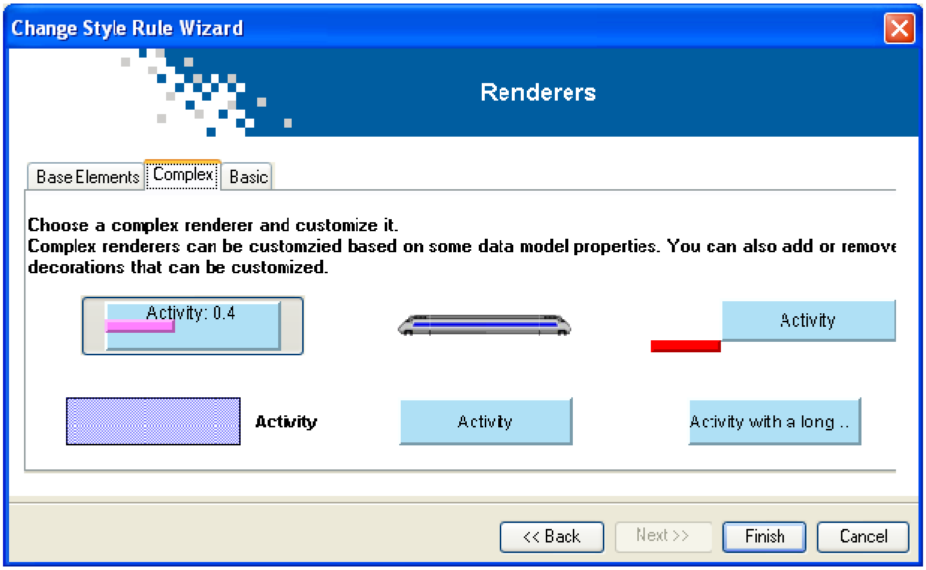
7. Click **Next** to opt to define a new renderer.

8. Select **New Renderer** and click **Next**.

9. Select a renderer.

   The Designer offers ready-made base elements for composite graphics, complex, and basic renderers. The summary renderer used to represent activities with an activity bar and start and end time symbols falls into the category of basic renderers. You are going to change it for one of the complex renderers shown in the following figure.

   Select the **Complex** tab.

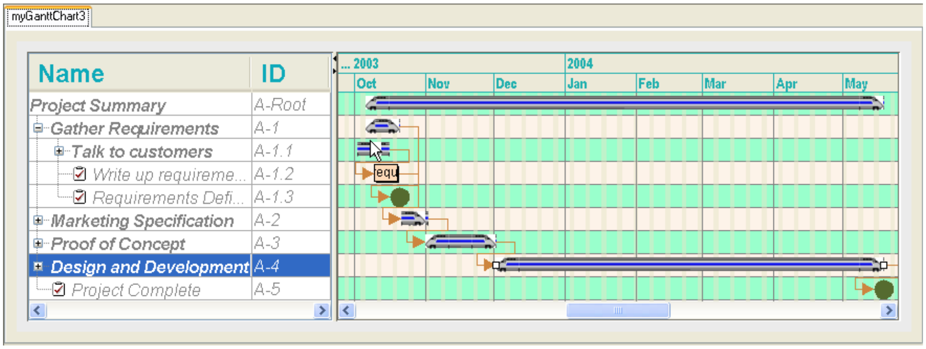


Choose a complex renderer and customize it.
Complex renderers can be customzied based on some data model properties. You can also add or remove decorations that can be customized.

Select **Activity bar with icons**.

10. Click **Finish**.

The following figure shows how the change in styling of parent activities is reflected immediately in the Gantt chart.



Name	ID
Project Summary	A-Root
Gather Requirements	A-1
Talk to customers	A-1.1
Write up requireme...	A-1.2
Requirements Defi...	A-1.3
Marketing Specification	A-2
Proof of Concept	A-3
Design and Development	A-4
Project Complete	A-5

Create style rule

In this section you will:

- Create new style rule conditions that apply to specific activities.
- Set the renderer for each style rule.
- Rename a style rule.
- Create a style rule for a constraint.

Create a style rule condition for specific parent activities

This style rule will have a more restricted application than the style rule `activity:parent`. The objective is to render parent activities started on or before October 6 2003 differently from other parent activities.

1. Select the style rule `activity:parent`.
2. Choose **Edit > Create Style Rule** or right-click and choose **Create Style Rule**.
3. Clear the option for displaying the Welcome page.
4. Click **Next** and select the **Data Model** tab.

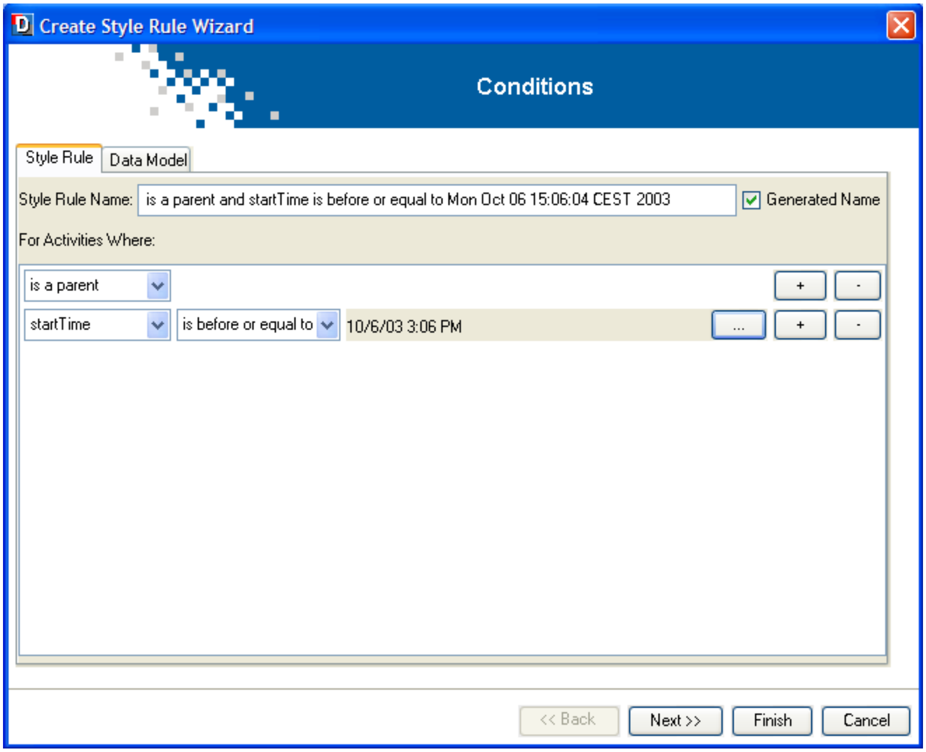
   You can see that there are two parent activities that start on October 6, 2003: A-1 Gather Requirements and A-1.1 Talk to customers.
5. Select the **Style Rule** tab to write the condition of the new style rule.

   You are going to write the new style rule by adapting the content of the style rule `activity:parent`. The objects affected by the rule are parent activities and they do not have any special properties of user-defined type.

   User-defined type is shown in the pink column in the data model view. The only activities shown to have this kind of property are leaf activities. (See *User-defined properties* for more information on user-defined types.)

   You need to specify the attribute that distinguishes the affected objects and causes the style rule to be applied selectively.
6. Click the plus sign and select **startTime**.
7. Select **is before or equal to**.
8. Click the ellipsis (**...)** to display the calendar that allows you to enter the date in the correct format.
9. Select **October 2003** and the date of the **6**th; click **Apply**.

   The following figure shows the completed condition.



Style Rule Name: is a parent and startTime is before or equal to Mon Oct 06 15:06:04 CEST 2003 ☑ Generated Name

For Activities Where:

is a parent

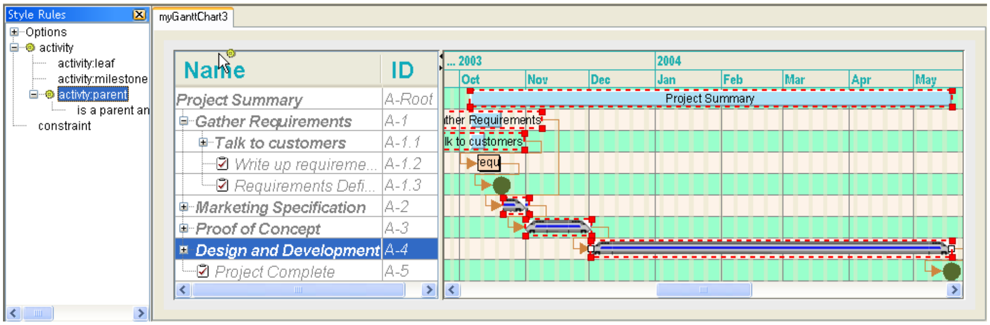
startTime | is before or equal to | 10/6/03 3:06 PM

Set the renderer for specific parent activities

You need to set the way the parent activities that satisfy the condition of the style rule will be rendered.

1. Click **Next** to set the renderer for the activities specified by the rule condition.
2. Select **New renderer** and click **Next** to define a different renderer for the specified activities.
3. Select the complex renderer **Activity bar with text: where the text is given by a property of the model** and click **Finish**.

The following figure shows that the effect of the new style rule on activities A-1 and A-1.1 is shown in the chart immediately.



Name	ID
Project Summary	A-Root
Gather Requirements	A-1
Talk to customers	A-1.1
Write up requireme...	A-1.2
Requirements Defi...	A-1.3
Marketing Specification	A-2
Proof of Concept	A-3
Design and Development	A-4
Project Complete	A-5

You can see that three style rules, as indicated by the yellow icons, apply to the objects selected in the chart:

- `activity`
- `activity:parent`
- `is a parent and startTime is before or equal to Monday Oct 06 15:06:04 CEST 2003`

The style rule lowest in the hierarchy is the rule with the most specific application.

Rename style rule

The default name of a new rule may not be useful to you. You can rename a style rule:

1. **Select** the style rule and **right-click**.
2. Choose **Rename**.
3. Type the new name of the rule and click **OK**.

Create a style rule condition for an activity with a user-defined type

This style rule is based on the style rule `activity:leaf`. The objective is to render activities that have the user-defined type `critical`, are the responsibility of the Engineering Department, and that are completed before January 20 2004. You might want to check the data model view to determine the likely scope of the rule. See *User-defined properties* for more information about user-defined types.

There are four activities marked as critical in the data model view. User-defined types are shown in the pink column of this view. Only one of these belongs to the Engineering Department. This activity, A-4.1 Phase 1 Development, has an end time of December 2, 2003.

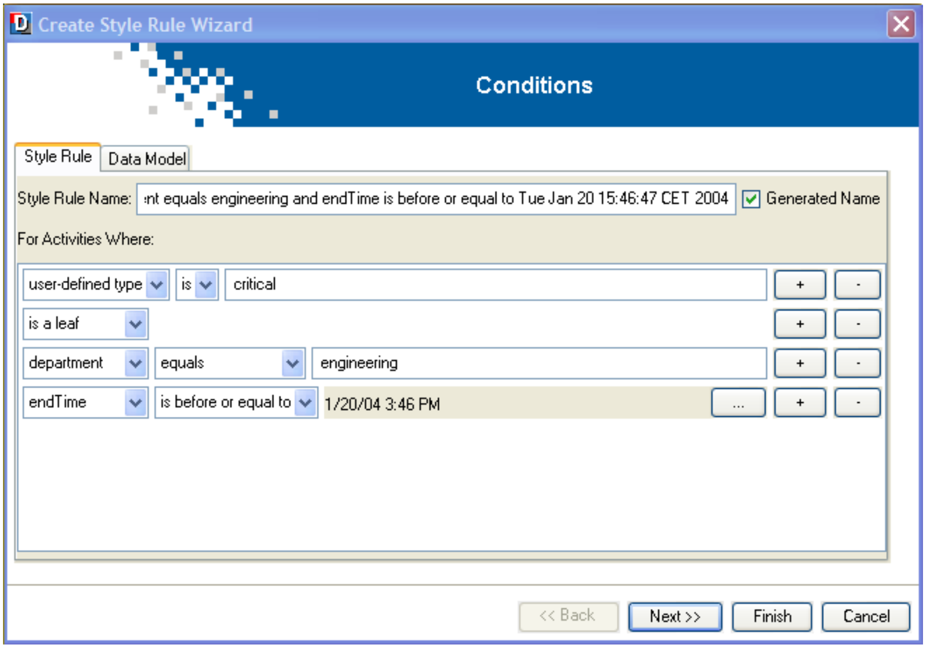
1. Select the style rule `activity:leaf`.
2. Choose **Edit > Create Style Rule**

   You are going to write the new style rule by adapting the content of the style rule `activity:leaf`. The objects affected by the rule are leaf activities with a user-defined type.

You need to specify the user-defined type and the attributes that distinguish the affected objects and cause the style rule to be applied selectively.

3. Select **user-defined type** and type **critical**.
4. Click the plus sign and select **is a leaf**.
5. Click the plus sign and select **department**.
6. Select **equals**.
7. Type **engineering**.
8. Click the plus sign to specify the next attribute.
9. Select **endTime**.
10. Select **is before or equal to**.
11. Click the ellipsis (**...**) to display the calendar that allows you to enter the date in the correct format.
12. Select **January 2004** and the date of the **20**th; click **Apply**.

    The following figure shows the completed condition.



Style Rule Name: :nt equals engineering and endTime is before or equal to Tue Jan 20 15:46:47 CET 2004 ☑ Generated Name

For Activities Where:

user-defined type | is | critical | + | -

is a leaf | + | -

department | equals | engineering | + | -

endTime | is before or equal to | 1/20/04 3:46 PM | ... | + | -

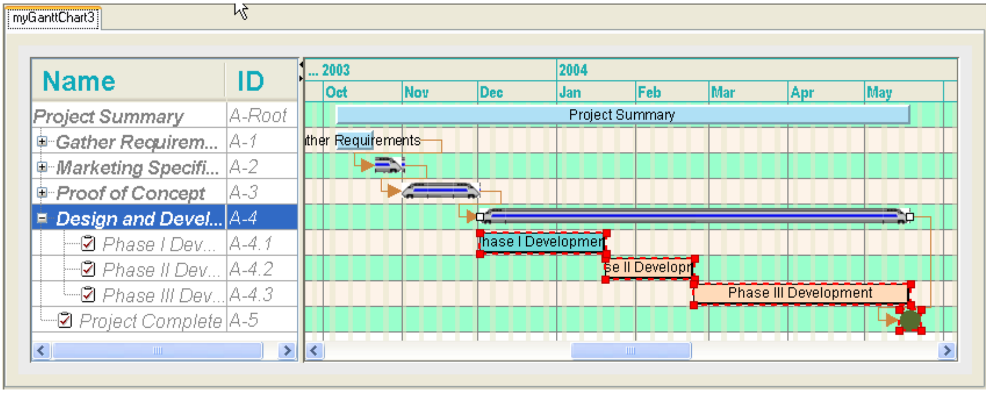
Set the renderer for the specified activity

Set the way the activity that satisfies the condition of the style rule will be rendered.

1. Click **Next** to set the renderer for the activity specified by the rule condition.

2. Select **New renderer** and click **Next** to define a different renderer for the specified activity.
3. Click the **Basic** tab to select a basic renderer for customization.
4. Choose the **Activity bar:simple activity bar** and click **Finish**.
5. Select the new style rule in the tree to customize the Activity bar.
6. In the **Styling Customizer**, click the **Label** tab and set **Color** to bright red (255,0,51).
7. Click **Apply**.
8. Click the **Shape** tab and set **Color** to mid-gray (153,153,153).
9. Click **Apply**.

The following figure shows the effects of this style rule.



Name	ID
Project Summary	A-Root
⊞ Gather Requirem...	A-1
⊞ Marketing Specifi...	A-2
⊞ Proof of Concept	A-3
⊟ Design and Devel...	A-4
☑ Phase I Dev...	A-4.1
☑ Phase II Dev...	A-4.2
☑ Phase III Dev...	A-4.3
☑ Project Complete	A-5

The new style rule is added to the tree after the rule `activity:parent`. Even though this rule applies to a leaf activity, it is too specific to be added to a part of the hierarchy that would branch directly under leaf activities. This is because the rule relates to objects with a user-defined type, which increases the specificity of the rule. This rule is so specific that you cannot move it up in the rule tree. Remember that the more specific a rule, the further down the hierarchy it appears.

When you select the new style rule, the yellow icons clearly show that the style rules for `activity` and `activity:leaf` can also apply, but are overridden by the new rule.

Rename the rule to `activity.critical:leaf.engineering.completion`.

Create a style rule condition for a constraint

This style rule is based on the style rule `constraint`. The objective is to render constraints of type **End-Start**.

1. Select the style rule `constraint`.
2. Choose **Edit > Create Style Rule**

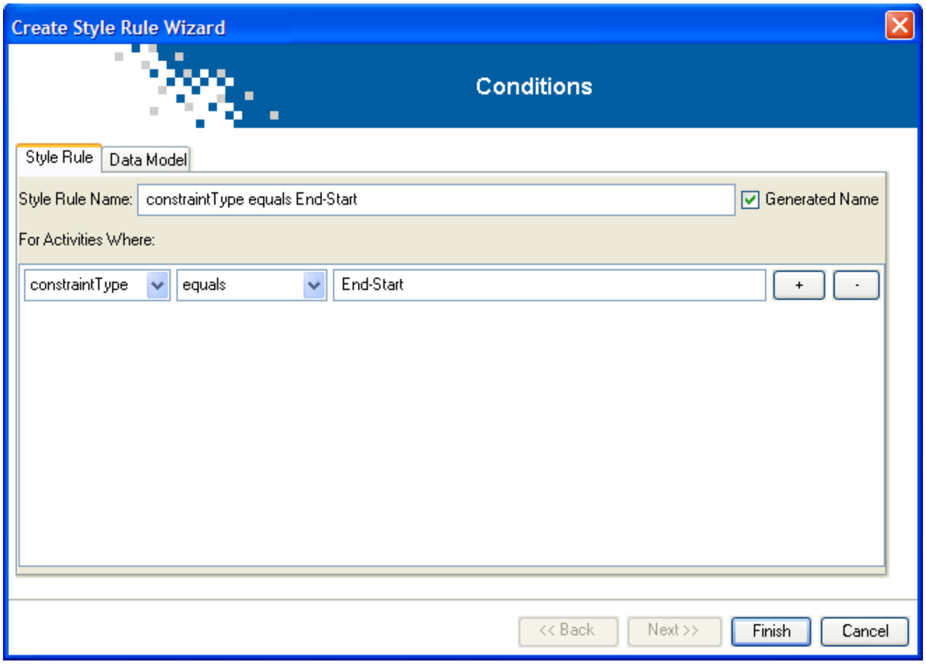
   You are going to write the new style rule by adapting the content of the style rule `constraint`.

You can specify the constraint type.

3. Select **equals**.

4. Type **End-Start**.

   The following figure shows the resulting style rule condition.



Style Rule Name: constraintType equals End-Start ☑ Generated Name

For Activities Where:

constraintType | equals | End-Start

5. Click **Finish**.

Set the renderer for the specified constraints

Set the way the constraints that satisfy the condition of the style rule will be rendered.

1. Select the new constraint.

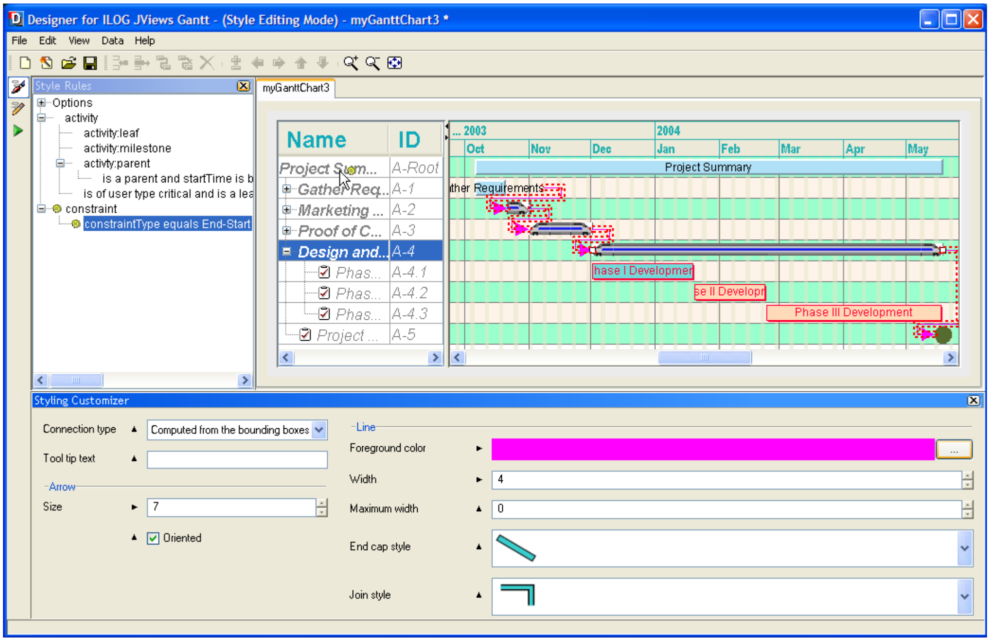
2. In the Styling Customizer, change the Foreground color of the line to magenta by selecting and applying the color identified as (**255,0,255**).

   Select this color from the Swatches; its tooltip identifies the desired value.

3. Increase the Line Width to **4**.

4. Reduce the Arrow Size to **5**.

The following figure shows the changed styling, which is restricted to constraints on activities of type End-Start. Note that the black triangles beside the changed rendering properties that are specific to this rule point to the right to show that these settings are local to the current rule. The black triangles beside the other properties point upwards to indicate that those properties are inherited from the generic constraint rule.



Designer for ILOG JViews Gantt - (Style Editing Mode) - myGanttChart3 *
File
Edit
View
Data
Help
Style Rules
myGanttChart3
Styling Customizer

- Options
- activity
  - activity:leaf
  - activity:milestone
  - activty:parent
    - is a parent and startTime is b
  - is of user type critical and is a lea
- constraint
  - constraintType equals End-Start

Name	ID
Project Sum...	A-Root
Gather Req...	A-1
Marketing ...	A-2
Proof of C...	A-3
Design and...	A-4
Phas...	A-4.1
Phas...	A-4.2
Phas...	A-4.3
Project ...	A-5

Connection type ▲ Computed from the bounding boxes

Tool tip text ▲

Arrow

Size ► 7

▲ ☑ Oriented

Line

Foreground color ►

Width ► 4

Maximum width ▲ 0

End cap style ▲

Join style ▲

Developing a resource-oriented Gantt chart

Explains how to create and style Gantt and Schedule charts.

In this section

Overview

A resource-oriented Gantt chart or Schedule chart shows resources along the y axis against time on the x axis. The resource table on the left of the chart displays information from the data model about the resources. There is one row for each resource. Each column in this table displays a property of the resource.

The Gantt sheet on the right of the chart displays reservations of the resources for activities. Each row in the Gantt sheet could contain reservation graphics to represent the activities for which the matching resource is reserved. The same resource can be reserved for more than one activity during the same time span. Therefore several reservation graphics could occupy the same horizontal area in the same row. The Designer manages the potential overlapping of reservation graphics.

Since one activity can reserve several resources and appear as several reservation graphics, constraints between activities are not displayed by default in the resource-oriented Gantt chart. Constraint links can be displayed if each activity reserves only one resource.

This type of chart is implemented by an instance of the class `IlvScheduleChart` in the Java API. See The Gantt beans in **Developing with the SDK**.

Creating a resource-oriented Gantt chart

Access the New Gantt Chart Wizard from the menu bar (**File > New from Wizard**) or by typing `Ctrl-W`.

In this section you will:

- Choose to create a new Gantt chart project from a template.
- Choose a Basic Schedule Chart as the template.

Starting from a template is the easiest way to create a Gantt chart.

Starting from a template

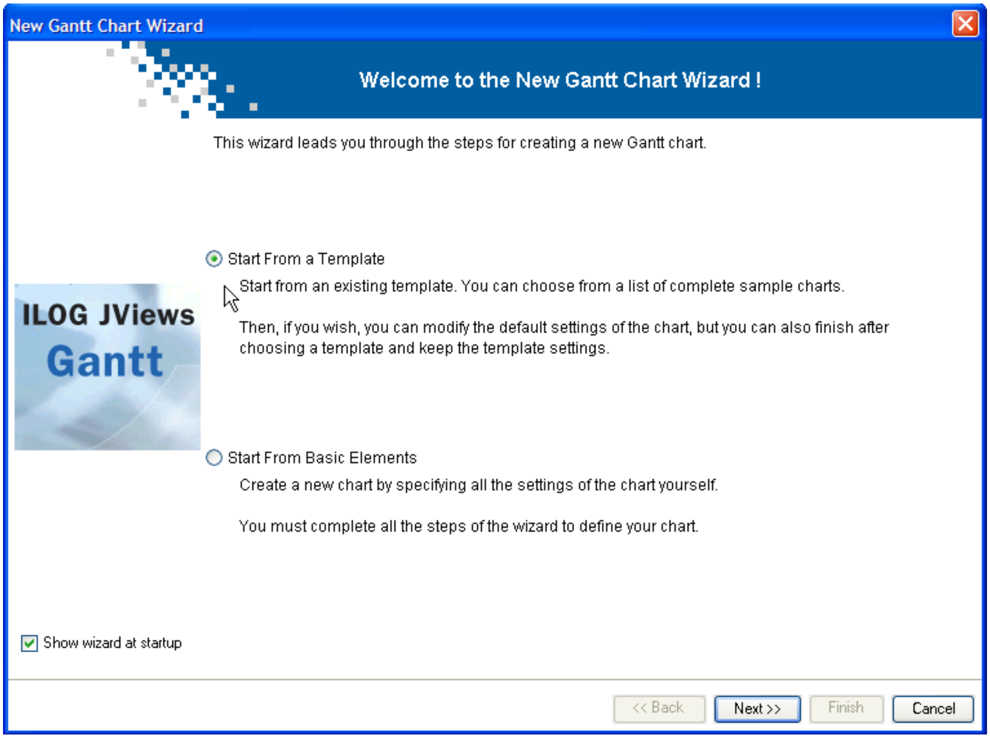
A template specifies the data source and the data for a chart as well as the styling of the chart and its data. There is nothing more to do in the New Gantt Chart Wizard after choosing the template. You can click **Finish** without having to pass through the other steps of the wizard. Using a template is a quick and easy way of getting a ready made Gantt chart.

1. Select the default **Start From a Template** by clicking **Next**.
2. Select the template **Basic Schedule Chart**.

   A preview of the resource-oriented Gantt chart is shown on the right, so you can see the data and the styling used in this template.
3. Click **Finish** to accept the data configuration and the styling as is.

You have a complete resource-based Gantt chart containing the data specified by the data source of the template project file. You can customize the styling of the chart and its data as you did for the activity-oriented Gantt chart.

The following figure shows that the same chart options are available for styling as in the activity-oriented Gantt chart.



New Gantt Chart Wizard

Welcome to the New Gantt Chart Wizard !

This wizard leads you through the steps for creating a new Gantt chart.

● Start From a Template

Start from an existing template. You can choose from a list of complete sample charts.

Then, if you wish, you can modify the default settings of the chart, but you can also finish after choosing a template and keep the template settings.

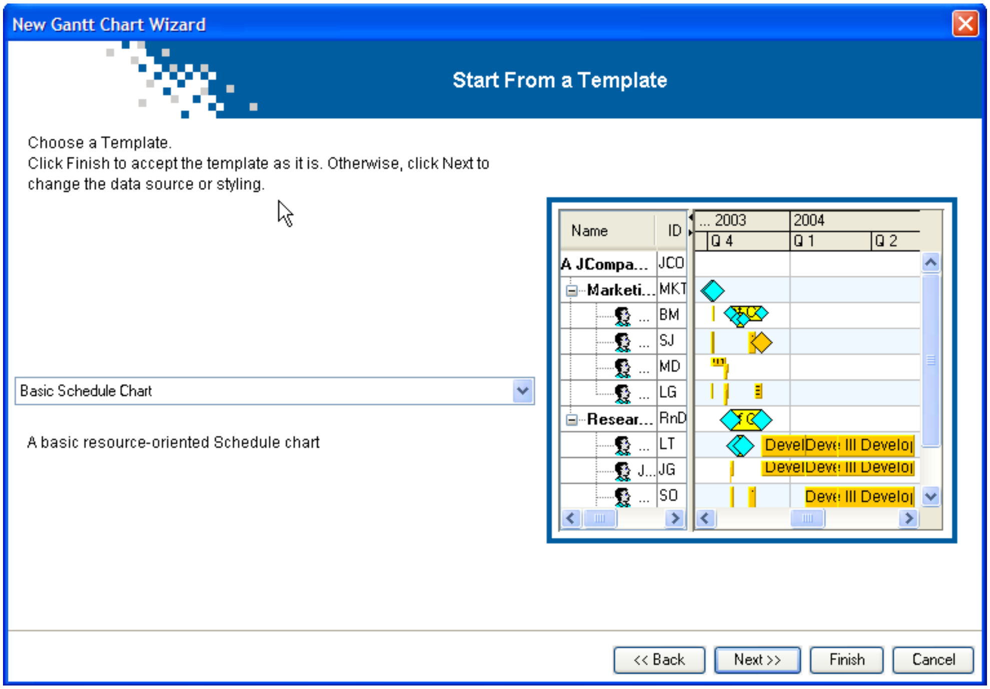
○ Start From Basic Elements

Create a new chart by specifying all the settings of the chart yourself.

You must complete all the steps of the wizard to define your chart.

☑ Show wizard at startup

The following figure shows how to select a template.


New Gantt Chart Wizard

Start From a Template

Choose a Template.
Click Finish to accept the template as it is. Otherwise, click Next to change the data source or styling.

Basic Schedule Chart

A basic resource-oriented Schedule chart

Styling data in a resource-oriented Gantt chart

There is no styling of resources. The styling of the reservations of a resource for one or more activities is based on activity style rules. See *Styling the data* for details of how to style activity bars.

Integrating your development into an application

Explains how to integrate a complete Gantt chart or a customized style sheet into an application.

In this section

Overview

You can integrate a complete Gantt chart 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.

Integrating a Gantt chart into an application

When you have saved your Gantt chart in the Designer, you have a project file that you can use to integrate the chart component into an application. There are sample project files in:

`<installdir>/jviews-gantt86/bin/designer/data/examples`

See also the gallery of Designer projects supplied in the JViews Gantt samples at:

`<installdir>/jviews-gantt86/samples/gallery`

To integrate a chart component into an application:

1. Make sure that all the project files are accessible from the application.

   The `.igpr`, `.css`, and `.xml` files must all be accessible from the application.

2. Create the Gantt chart instance and load the project file into the application.

   Creating the Gantt chart instance and loading the project file shows you how to create an instance of a Gantt chart, which can be an instance of `IlvGanttChart` as in this example or of `IlvScheduleChart`, and load the project file into the application.

   Creating the Gantt chart instance and loading the project file

   ```
   IlvGanttChart chart = new IlvGanttChart();
   try {
        URL project = ...;
        // Apply the project file to the chart
           chart.setProject(project);
        } catch (IOException e) {
        ...;
        } catch (IlvStylingException se) {
        ...;
         }
   ```

   The Gantt chart is thus configured according to the settings of the project file.

   The project file contains the type of chart it was saved for, which can be an activity-oriented Gantt chart (`IlvGanttChart`) or a resource-oriented Gantt chart (`IlvScheduleChart`).

   The code for creating the Gantt chart instance and loading the project file is shown in:

   `<installdir>/jviews-gantt86/codefragments/application/loadProject/src/LoadProject.java`

   When you run this example it displays buttons that allow you to choose the type of chart developed in the Designer. You must then select the path to your chart. If the type of chart used in the project does not correspond to the type of chart you have chosen to load, you will receive an error message.

Note: A project file includes both the data and the style sheet. When a project is set, it replaces any data source and style sheets that may already be set on the chart.

This code allows you to integrate your Gantt chart into an application, for example, a Swing application. You still need to write the code for using the Gantt chart in the application. JViews Gantt does not provide any special facilities for creating menus or toolbars, for example. You need to develop such features in your application for yourself. See *Writing an application* for more information.

Integrating a style sheet into an application

If you have a custom data model and your application is already defined, you can use the Designer for styling only on a sample of dummy data or on an XML dump of your data. Then you can integrate the style sheet from the Designer into your application. You can apply the styling defined in the style sheet to the real data used in the application. The Designer makes it much easier to write the style sheet.

Follow these steps to style the chart component of your application using an existing project style.

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 chart and loading the data model, use the code shown in *Loading a style sheet Into an application*:

   Loading a style sheet Into an application

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

See Applying styles in **Developing with the SDK** for more information on how to load style sheets.

You can also look at the Gantt and Schedule chart CSS samples in:

`<installdir>/jviews-gantt86/samples/cssGantt`

`<installdir>/jviews-gantt86/samples/cssSchedule`

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 and 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 Gantt chart configured by the project file opens automatically in the Designer. The drag-and-drop action is available in both Style Editing and Gantt Chart Editing modes.

Getting to know the Designer

Introduces you to the different types of chart and the subcomponents that are used to build them.

In this section

Working in different modes
Describes the modes available in the Designer.

Style Rules
Explains how style rules are used to style your chart and data rules to style application data.

Using wizards
Describes the wizards supplied by the Designer and how to use them.

The Rule Menu
Explains how to use the rule management commands in the Rule Menu.

The Undo and Redo facilities
Explains how to undo and redo an action.

Checking rules
Explains how to use the Check Rules facility to validate your work.

Messages for troubleshooting
Describes the message details shown in the Messages panel.

Printing
Describes the printing facilities available in the Designer.

Working in different modes

Describes the modes available in the Designer.

In this section

Overview
Describes the modes available in the Designer.

Style Editing Mode
Explains how you use the Gantt Chart Editing Mode to modify a chart and the underlying data model.

Gantt Chart Editing Mode
Explains how you use the Gantt Chart Editing Mode to modify a chart and the underlying data model.

Preview Mode
Explains how you use Preview mode to interact with a chart.

Overview

You can work in the following modes:

- *Style Editing Mode*
- *Gantt Chart Editing Mode*
- *Preview Mode*

There are specific default interactions associated with each mode. In all modes you can zoom in and out.

Style Editing Mode

This icon in the vertical toolbar represents Style Editing Mode. In this mode you can select a rule under Style Rules or select a chart component or data object in the chart and specify the styling.

You can customize the styling of chart components through their associated customizers. See *Customizing Gantt chart Options*.

You can style data by writing style rules. These rules are written by creating or changing style rule conditions and associating them with a renderer. You can do this in the Create Style Rule Wizard or the Change Style Rule Wizard. See *Create style rule* and *Change style rule*. The condition represents the expressions in the left part (the selector) of a CSS style rule. The rendering represents the right part of a CSS style rule.

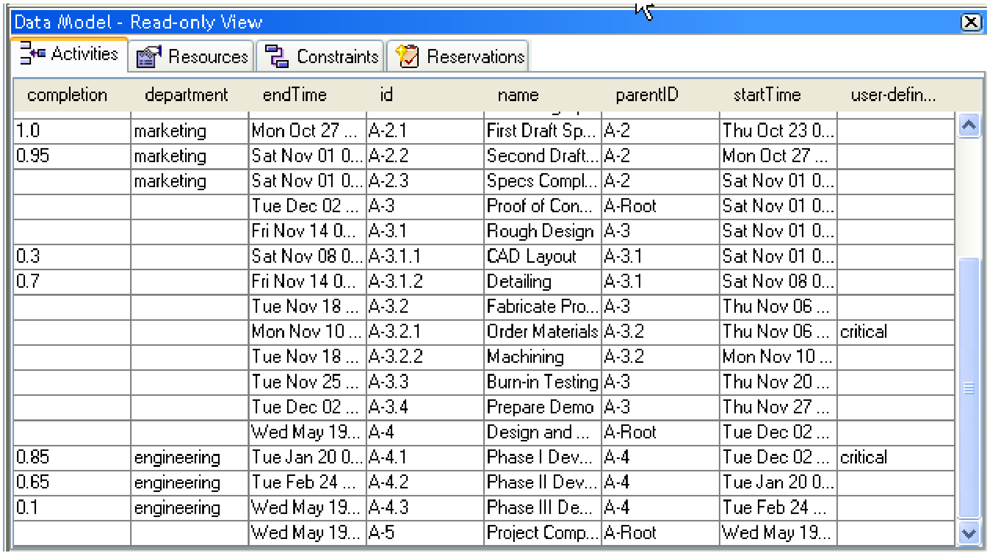
The style rules define the styling used in the style sheet which is the CSS file used by a project in the Designer.

When you write style rules, you can choose to apply the renderer inherited from a more generic related rule higher in the style rule hierarchy or you can choose a new renderer that applies only to the more specific context defined by the style rule condition. You can define the new renderer by choosing one of the default renderers supplied in the Designer and customizing it in the Styling Customizer.

Viewing the data model in Style Editing Mode

You can display a read-only view of the data model by choosing **View > Data Model**. When the Data Model option is checked the Data Model — Read-only View is displayed by default in a pane below the Gantt chart. This panel is dockable in any position within the main panel or can be a separate window.

The following figure shows Read-only view of data model in Style Editing Mode.



completion	department	endTime	id	name	parentID	startTime	user-defin...
1.0	marketing	Mon Oct 27 ...	A-2.1	First Draft Sp...	A-2	Thu Oct 23 0...	
0.95	marketing	Sat Nov 01 0...	A-2.2	Second Draft...	A-2	Mon Oct 27 ...	
	marketing	Sat Nov 01 0...	A-2.3	Specs Compl...	A-2	Sat Nov 01 0...	
		Tue Dec 02 ...	A-3	Proof of Con...	A-Root	Sat Nov 01 0...	
		Fri Nov 14 0...	A-3.1	Rough Design	A-3	Sat Nov 01 0...	
0.3		Sat Nov 08 0...	A-3.1.1	CAD Layout	A-3.1	Sat Nov 01 0...	
0.7		Fri Nov 14 0...	A-3.1.2	Detailing	A-3.1	Sat Nov 08 0...	
		Tue Nov 18 ...	A-3.2	Fabricate Pro...	A-3	Thu Nov 06 ...	
		Mon Nov 10 ...	A-3.2.1	Order Materials	A-3.2	Thu Nov 06 ...	critical
		Tue Nov 18 ...	A-3.2.2	Machining	A-3.2	Mon Nov 10 ...	
		Tue Nov 25 ...	A-3.3	Burn-in Testing	A-3	Thu Nov 20 ...	
		Tue Dec 02 ...	A-3.4	Prepare Demo	A-3	Thu Nov 27 ...	
		Wed May 19...	A-4	Design and ...	A-Root	Tue Dec 02 ...	
0.85	engineering	Tue Jan 20 0...	A-4.1	Phase I Dev...	A-4	Tue Dec 02 ...	critical
0.65	engineering	Tue Feb 24 ...	A-4.2	Phase II Dev...	A-4	Tue Jan 20 0...	
0.1	engineering	Wed May 19...	A-4.3	Phase III De...	A-4	Tue Feb 24 ...	
		Wed May 19...	A-5	Project Comp...	A-Root	Wed May 19...	

Use the tabs to view the different object types (Activities, Resources, Constraints, and Reservations) in the data model.

Gantt Chart Editing Mode

This icon in the vertical toolbar represents Gantt Chart Editing Mode. In this mode you can modify the chart and the underlying data model. You can insert and delete activities or resources and create constraints or reservations. You can indent and outdent data objects and move them up or down in the hierarchy.

In this mode, you cannot change the styling of objects.

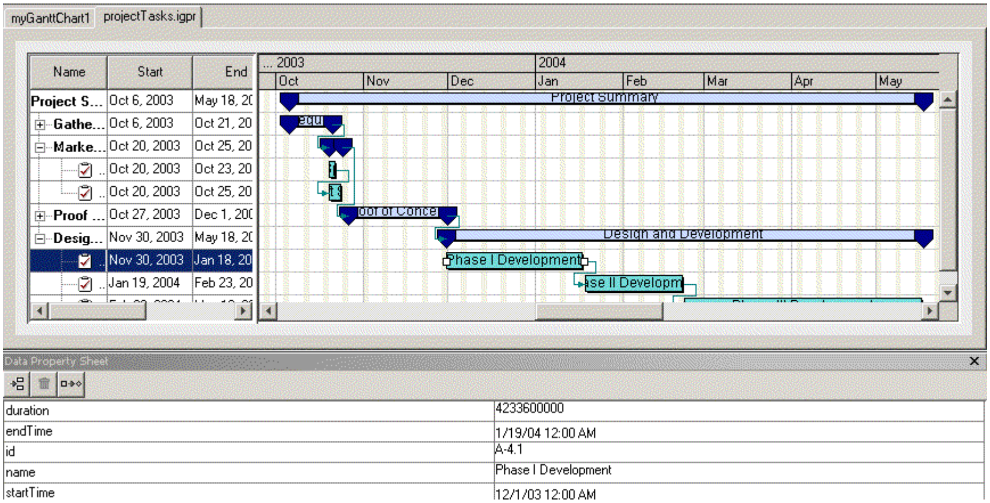
In this mode, you can expand the data nodes and scroll vertically and horizontally, so that all nodes can be visible. (See expanded data node and visible data node.) It is easier to see leaf activities or resources in this mode than in Style Editing Mode.

Selecting an object in the Gantt Sheet (activities or constraints in a Gantt chart; resources or reservations in a Schedule chart) displays the properties of this object in the Data Property Sheet. The following figure shows the activity Phase 1 Development selected in the Gantt Sheet (upper right pane) and the properties of this activity in the Data Property Sheet (lower pane).

The Data Property Sheet is also visible in Style Editing Mode.

You can then remove, add, or modify some of the properties.

Note: Note that some properties are not editable.



Name	Start	End
Project S...	Oct 6, 2003	May 18, 20
⊞ Gathe...	Oct 6, 2003	Oct 21, 20
⊟ Marke...	Oct 20, 2003	Oct 25, 20
..	Oct 20, 2003	Oct 23, 20
..	Oct 20, 2003	Oct 25, 20
⊞ Proof ...	Oct 27, 2003	Dec 1, 200
⊟ Desig...	Nov 30, 2003	May 18, 20
..	Nov 30, 2003	Jan 18, 20
..	Jan 19, 2004	Feb 23, 20

duration	4233600000
endTime	1/19/04 12:00 AM
id	A-4.1
name	Phase I Development
startTime	12/1/03 12:00 AM

Adding a user-defined type

To add a user-defined type:

1. Select an object.

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

   If you want to enter multiple values, separate each value in the list by a comma.

User-defined types are identified by the value of the `tags` property. See *User-defined properties*.

Adding a data property

To add a new property to an object:

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

   Values can only be primitive values.

Removing a property

To remove a property of an object:

1. Select the property in the **Data Property Sheet**.
2. Click the **Delete Property** button in the toolbar.

Preview Mode

This icon in the vertical toolbar represents Preview Mode. In this mode you can interact with the chart and its objects using the interactors zoom in, zoom out, or zoom to fit to see the dynamic effects your style rules will have in your application.

In this mode, you cannot change the styling of objects or the data in the chart.

Style Rules

Explains how style rules are used to style your chart and data rules to style application data.

In this section

Overview
Describes the options available in the Style Rules pane and how style rules are saved.

Options
Explains how you use the Options section to style a chart and its individual subcomponents.

Data rules
Explains what data rules are and where to find them in the Designer.

Styling Customizers
Explains what the Styling Customizer is and the editing options available for the different properties.

Styling Properties
Explains how properties are displayed in the Styling Properties list.

Priority
Explains how rule priority is calculated.

Debugging the style sheet
Explains how to investigate the effect of rule priority over style change .

Resetting default property values
Describes the effect of inherited default values on a renderer.

Overview

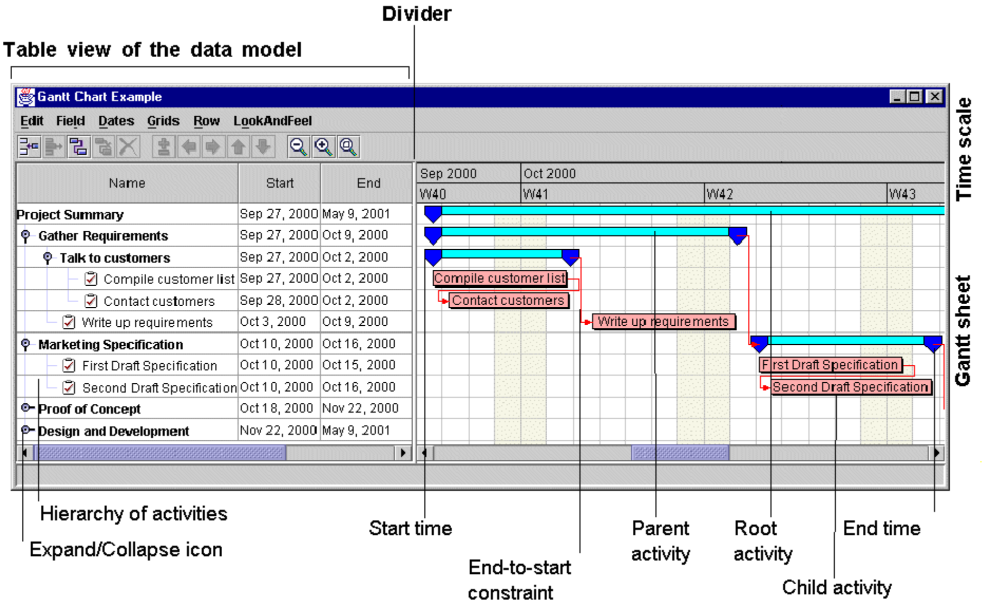
The Style Rules pane displays the Options for styling your chart as a whole and data rules for styling the application data. When you save a chart as a project, the style rules (options and data rules) are saved in a style sheet (`.css` file) with the CSS for Java syntax. For details of the syntax, see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

Options

The Options section of the Style Rules pane allows you to style the chart at the level of the chart as a whole and of its individual subcomponents. Each chart you work on has its own style sheet and therefore its own options.

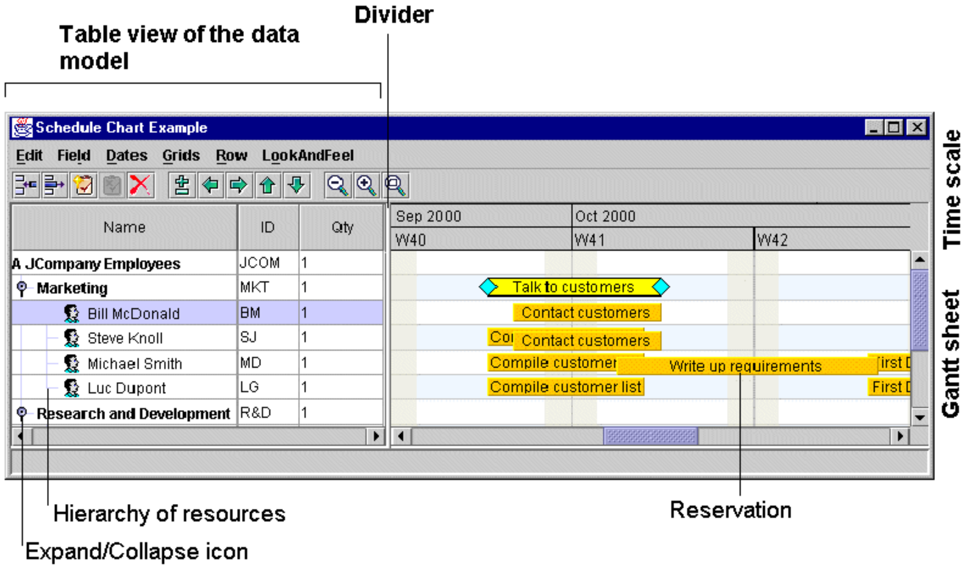
There are Options for the chart component and its subcomponents. Options are controlled through properties in the Styling Customizer for the chart component or each of its subcomponents. The Styling Customizer appears in the lower pane of the Designer when you select the option in the Style Rules pane.

The following figure shows the different parts of a Gantt chart.



Name	Start	End
Project Summary	Sep 27, 2000	May 9, 2001
Gather Requirements	Sep 27, 2000	Oct 9, 2000
Talk to customers	Sep 27, 2000	Oct 2, 2000
Compile customer list	Sep 27, 2000	Oct 2, 2000
Contact customers	Sep 28, 2000	Oct 2, 2000
Write up requirements	Oct 3, 2000	Oct 9, 2000
Marketing Specification	Oct 10, 2000	Oct 16, 2000
First Draft Specification	Oct 10, 2000	Oct 15, 2000
Second Draft Specification	Oct 10, 2000	Oct 16, 2000
Proof of Concept	Oct 18, 2000	Nov 22, 2000
Design and Development	Nov 22, 2000	May 9, 2001

The following figure shows the different parts of a Schedule chart.



Name	ID	Qty
A JCompany Employees	JCOM	1
Marketing	MKT	1
Bill McDonald	BM	1
Steve Knoll	SJ	1
Michael Smith	MD	1
Luc Dupont	LG	1
Research and Development	R&D	1

You can customize the styling of the Gantt chart through options that apply to:

♦ The chart itself

You can customize the appearance of the chart through properties that control:

- Border
- Inside border
- Row height
- Visibility of the table and the sheet

You can customize the appearance of the divider through properties that control:

- Location
- Size
- Move mode

You can customize the appearance of the scrollbars through properties that control:

- Horizontal scrollbar visibility
- Vertical scrollbar policy

You can customize time and duration through properties that control:

- Beginning of the visible time interval
- Animation steps of the visible time interval
- Unbounded time interval scrolling mode

    - Bounded time interval scrolling mode

    You can customize the colors and fonts of the chart, Gantt sheet, table, table header, and time scale through properties that control:

    - Background color
    - Foreground color
    - Font

- The table

    You can customize the appearance of the table and table header through properties that control:

    - Background color
    - Foreground color
    - Font

    You can customize the appearance of the grid through a property that controls:

    - Color

    You can customize the columns of the table through properties that control their appearance:

    - Margin
    - Selection allowed

    You can customize the columns of the table through a property that uses the column names to control:

    - Column order

- The sheet

    You can customize the appearance of the Gantt sheet through properties that control:

    - Background color
    - Background pattern location
    - Constraint layer visible
    - Displaying constraints

    The Control group of properties of the Gantt sheet let you control whether:

    - Tooltips are enabled
    - Activity graphics are drawn over multiple rows

    and whether parent activities are:

    - Editable
    - Movable

    - Refreshed when changing from leaf to parent or vice versa

  and whether milestones are:

    - Refreshed when the activity time interval changes from null to non-null or vice versa

  The Advanced group of properties of the Gantt sheet let you control:

    - Autofit to contents
    - Antialiasing
    - Double buffering
    - Optimized translation

- The time scale

  You can customize the time scale through properties that control:

    - Background color
    - Foreground color
    - Font

- The horizontal grid

  You can customize the lines of the horizontal grid through a property that controls:

    - Color

  You can customize the rows of the horizontal grid through properties that control:

    - Fill
    - Even row color
    - Odd row color

- The vertical grid

  You can customize the colors of the vertictal grid through a property that controls:

    - Foreground color

You can customize the weekend grid through properties that control:

- Display of weekends
- Color of weekends
- Printing of weekends as opaque

You can use all the properties listed for Options in this section directly in the Designer. To control styling through other properties, you must use the SDK. See Styling Gantt and Schedule chart components in *Developing with the SDK*.

The following table shows the components and their corresponding CSS element type and ID together with their properties available in the Designer and the property type. See also *Styling Properties*.

Option	CSS Element Type and ID	Designer Property Fields	Type

The Gantt or Schedule chart component	`chart`	Border
		Inside border
		Row height
		Table visible
		Gantt sheet visible
		Divider location
		Divider size
		Divider opaque move mode
		Horizontal scrollbar visible
		Vertical scrollbar policy
		Beginning of the visible time interval
		Animation steps of the visible time interval
		Unbounded time interval scrolling mode
		(`minVisibleTime` is `null`
		and `maxVisibleTime` is `null)`
		Bounded time interval scrolling mode
		(`minVisibleTime` is not `null`
		and `maxVisibleTime` is not `null)`
		Background color
		Foreground color
		Font
			`Border`
			`Border`
			`int`
			`boolean`
			`boolean`
			`int`
			`int`
			`boolean`
			`boolean`
			`int`
			`Date`
			`int`
			`Date`
			`Date`
			`Color`
			`Color`
			`Font`

The Gantt table	`table`	Background color
		Foreground color
		Font
		Grid color
		Column margin
		Selection allowed
		Column order
			`Color`
			`Color`
			`Font`
			Color
			`int`
			`boolean`
			`String`

The Gantt sheet	`sheet`	Background color
		Background pattern location
		Constraint layer visible
			`Color`
			URL
			`boolean`

Option	CSS Element Type and ID	Designer Property Fields	Type
		Displaying constraints Tooltips enabled Activity graphics drawn over multiple rows Parent activity editable Parent activity movable Refresh parent activity when changing from leaf to parent or vice versa Refresh milestone when activity time interval changes from null to non-null or vice versa Autofit to contents Antialiasing Double buffering Optimized translation	boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean
The time scale	timeScale	Background color Foreground color Font	Color Color Font
The horizontal grid of the Gantt sheet	horizontalGrid	Line color Rows filled Even row background color Odd row background color	Color boolean Color Color
The vertical grid of the Gantt sheet	verticalGrid	Foreground color Display weekends Color of weekends Print weekends as opaque	Color boolean Color boolean

Data rules

A data rule is a style rule that applies to the data in the chart. It 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 renderers used to visualize the data in the chart.

In the Designer, you see the data rules 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.

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 renderer 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 checkbox 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 calendar system in which to select date and time
- 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 inherited:

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

Click the arrow icon to toggle the status.

Important: 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.

Debugging the style sheet

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 default property values

A renderer 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 default values of the renderer. 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 renderer.

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.

Using wizards

Describes the wizards supplied by the Designer and how to use them.

In this section

Wizards in the Designer

The Designer provides the following wizards to guide you through the creation of charts and style rules:

- *New Gantt Chart Wizard*
- *Create Style Rule Wizard*
- *Change Style Rule Wizard*
- *Reconfigure Data Wizard*

New Gantt Chart Wizard

To open the New Gantt Chart 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 Gantt Chart Wizard allows you to specify the data on which a chart is to be based and the initial look of the chart. You can then refine the styling by working in Style Editing Mode, see *Style Editing Mode*.

The New Gantt Chart Wizard has the following pages:

- Welcome, with choice between template and basic elements.
- Choose a Template, with a list of the available templates (predefined or user-defined).

  This page only appears if you have chosen the template option and not the basic elements option.
- Choose the Type of Data Source, with a list of the supported data source types.
- Gantt Chart Data Connection, with browse facilities to load from a file or database.
- Define Mapping from Tables to Gantt Chart Model, with a preview of the property-column mapping and editing facilities.
- Select the Type of Chart, with a choice between resource-oriented chart (Schedule chart) or activity-oriented chart.
- Choose a Theme, with a choice of color schemes.
- Exit.

Change Style Rule Wizard

To open the Change Style Rule Wizard, choose **Edit > Change Style Rule** or select the data rule that you want to change, right-click, and choose **Change Style Rule**.

The Change Style Rule Wizard simplifies the changing of the conditions and renderers in data rules. The pages are the same as those in the Create Style Rule Wizard, see *Create Style Rule Wizard*.

Create Style Rule Wizard

To open the Create Style Rule Wizard, choose **Edit > Create Style Rule** or select a data rule, right-click, and choose **Create Style Rule**.

The Create Style Rule Wizard simplifies the definition of data rules which can include attribute conditions. It also allows you to select a graphic representation.

Data rules

In the Create Style Rule Wizard, the items you can select are as follows:

- object: the type of object to which the rule will apply.
- user-defined type: the object subtype to which the rule will apply, according to the types you have defined
- state: the runtime state to which the rule will apply: `parent`, `leaf`, `milestone`, or `selected`

  Note that `parent` and `leaf` are only relevant to activities and resources and that a `milestone` is a zero-duration activity.
- attribute: the name of an attribute of the object in your data model for which a specific value is tested

The plus signs allow you to add further states and attribute conditions to which the rule applies.

The object types are activities and constraints. You may also have defined user-defined types, for example, `critical` for a subset of activities. You can make further selections (state or attribute or both) to set a more restrictive condition.

Attribute conditions

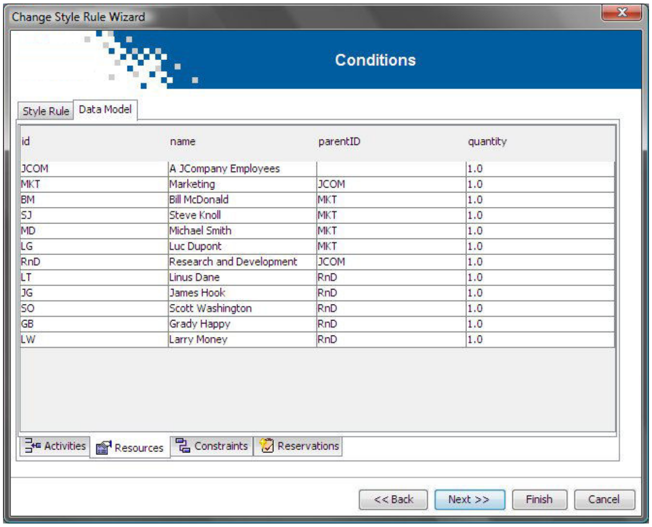
An attribute condition tests one or more attributes against a value. If you select an attribute, you must then set a condition on its value by selecting an operator from the operator menu and entering a value. You can set several attribute conditions.

The operator menu reflects the type of the attribute: integer, real, string, or date. For integer or real attributes, the operator menu offers exists and is not null, is greater than, is less than, is greater than or equal to, is less than or equal to, equals, does not equal. For string attributes, the operator menu offers exists and is not null, equals, does not equal, contains. For dates, the operator menu offers exists and is not null, is after, is before, is after or equal to, is before or equal to, equals, does not equal.

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 Data Model tab of the Conditions page. The table shows the property name and the current value for each property, on separate tabs for Activities, Resources, Constraints, and Reservations.

The following figure shows a table view of the data model for resources in a Style Rule Wizard.



id	name	parentID	quantity
JCOM	A JCompany Employees		1.0
MKT	Marketing	JCOM	1.0
BM	Bill McDonald	MKT	1.0
SJ	Steve Knoll	MKT	1.0
MD	Michael Smith	MKT	1.0
LG	Luc Dupont	MKT	1.0
RnD	Research and Development	JCOM	1.0
LT	Linus Dane	RnD	1.0
JG	James Hook	RnD	1.0
SO	Scott Washington	RnD	1.0
GB	Grady Happy	RnD	1.0
LW	Larry Money	RnD	1.0

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 Data Model tab.

Renderer mapping

Each type of data in a chart is represented by a renderer, for example, an Activity renderer for activities. In the wizard, you can accept the inherited renderer for the type of data or decide to choose a different renderer.

Renderers

A renderer can display a shape, an image (GIF, JPG, PNG or SVG), a text, or combinations of these (composite graphics).

Some predefined renderers are supplied in palettes in the Renderers page of the wizard.

Reconfigure Data Wizard

To open the Reconfigure Data Wizard, choose **Data > Reconfigure Data**.

The Reconfigure Data Wizard essentially allows you to revisit the parts of the New Gantt Chart Wizard that concern the data. You can change to a different type of data source (XML file, flat file, database, or in-memory) or keep the same data source but load data from a different file or database. You can also change the query used to load data from a database, for example, to add extra columns.

The Rule Menu

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

To create a style rule.

- Right-click a data rule, and choose **Create Style Rule Wizard** on the Rule Menu.

  See *Create Style Rule Wizard* for more information.

To change a style rule

- Right-click a data rule, and choose **Change Style Rule Wizard** on the Rule Menu.

  See *Change Style Rule Wizard*.

To cut a style rule:

- Right-click a style rule, and 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*.

To copy a style rule:

- Right-click a style rule, and choose **Copy** on the Rule Menu.

  Once you have copied a rule, you can do a Paste or Paste Styling Properties action.

To paste a style rule:

- Right-click the target style rule and choose **Paste** on the Rule Menu.

  This action changes both the conditions and the styling properties of the target rule.

Note: Before pasting, you must have already copied the style rule to be pasted.

To paste styling properties:

- Right-click the target style rule and choose **Paste Styling Properties** on the Rule Menu.

  This action leaves the conditions of the target rule unchanged and changes the styling properties.

Note: Before pasting, you must have already copied the style rule to be pasted.

To delete a style rule:

- ♦ Right-click the target style rule and choose **Delete** on the Rule Menu.

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

To customize the appearance of a style rule by renaming:

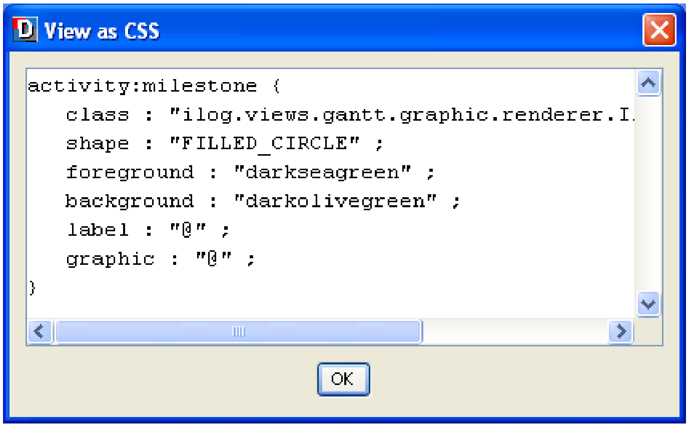
1. Right-click the target style rule and choose **Rename** on the Rule Menu.
2. Enter your preferred name.

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

To see the CSS code for a style rule:

1. Right-click the target style rule and choose **View** on the Rule Menu.

   The View as CSS dialog appears.



```
activity:milestone {
   class : "ilog.views.gantt.graphic.renderer.I.
   shape : "FILLED_CIRCLE" ;
   foreground : "darkseagreen" ;
   background : "darkolivegreen" ;
   label : "@" ;
   graphic : "@" ;
}
```

2. Click **OK** to close the dialog.

   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.

To reorder the style rules within a subtree:

1. A style rule that follows others may override property settings made by the preceding style rules. The order therefore represents a decision on priority.

   To move a style rule up:

   1. Right-click the target style rule and choose **Move up** on the Rule Menu.

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

You cannot place a less specific rule below a more specific rule.

2. To move a style rule down:

   1. Right-click the target style rule and choose **Move down** on the Rule Menu.

> **Note**: Overriding only occurs if the same property is set. If two rules set different properties, their order is unimportant.
>
> You cannot place a less specific rule below a more specific rule.

By default, all style rules defined are enabled. To enable then disable a style rule:

1. Right-click the target rule and 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.

2. Right-click the target rule and choose **Enable** on the Rule Menu.

   When you save a project file the enabled or disabled state of the rules is also saved.

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

There are unlimited levels of Undo.

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

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

To undo and redo an action:

1. Undo the action by selecting **Edit > Undo**.
2. Redo the action by selecting **Edit > Redo**.

Checking rules

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.)

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.

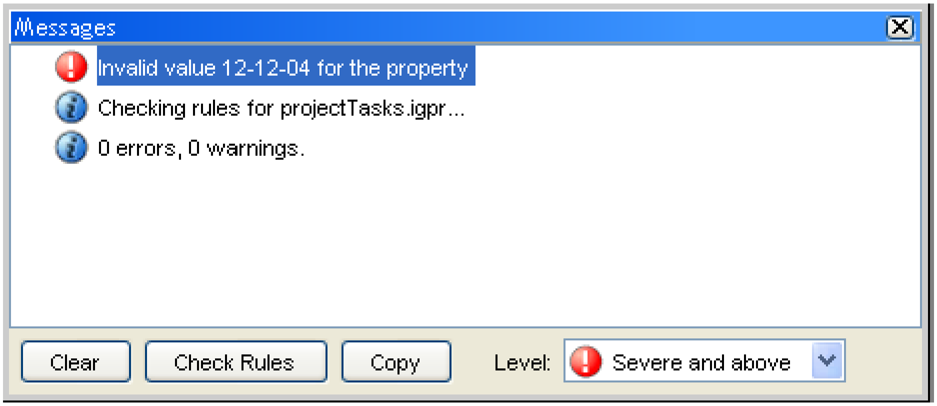
To check your rules:

- Do one of the following:
  - Click **Edit > Check Rules**.
  - Click the Check Rules button 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 following figure shows the Messages panel displaying a list of messages.



Invalid value 12-12-04 for the property

Checking rules for projectTasks.igpr...

0 errors, 0 warnings.

Level: Severe and above

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 objects in the chart (if any)
- The property setting in the Styling Customizer (if the cause was a property setting)

Printing

Describes the printing facilities available in the Designer.

In this section

Overview

In the Designer, you have the following printing facilities for a Gantt chart:

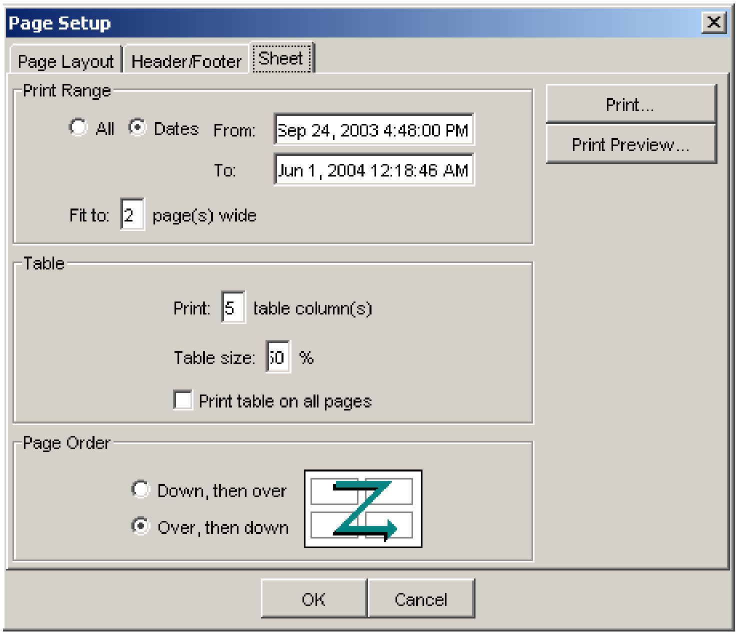
- *Page Setup*
- *Print Preview*
- *Print*

Print operations are available from the **File** menu.

Page Setup

You can access Page Setup by choosing **File > Page Setup** or by selecting the Setup tab in Print Preview. The Page Setup window offers extensive setup facilities.

The following figure shows the Page setup property sheet.



Print Range

○ All ● Dates From: Sep 24, 2003 4:48:00 PM

To: Jun 1, 2004 12:18:46 AM

Fit to: 2 page(s) wide

Table

Print: 5 table column(s)

Table size: 50 %

☐ Print table on all pages

Page Order

○ Down, then over

● Over, then down

Page Layout

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 chart.

Header/Footer

The Header/Footer tab gives you a selection of headers and footers to choose from and displays a preview of your selection. You can define a custom header and footer. A custom header or footer allows you to set a specific font and to insert and position page numbers, date, time, and file name.

Sheet

The fields in the Sheet tab allow you to control the print range, the way the table is processed, and the order in which pages are printed.

The following table describes the fields that you can set to control the way a Gantt chart is printed. The values in these fields determine the printed data window, which part of the Gantt chart is printed, how the Gantt chart fits on the page, and so on.

The following tale lists the fields used to control Page Setup.

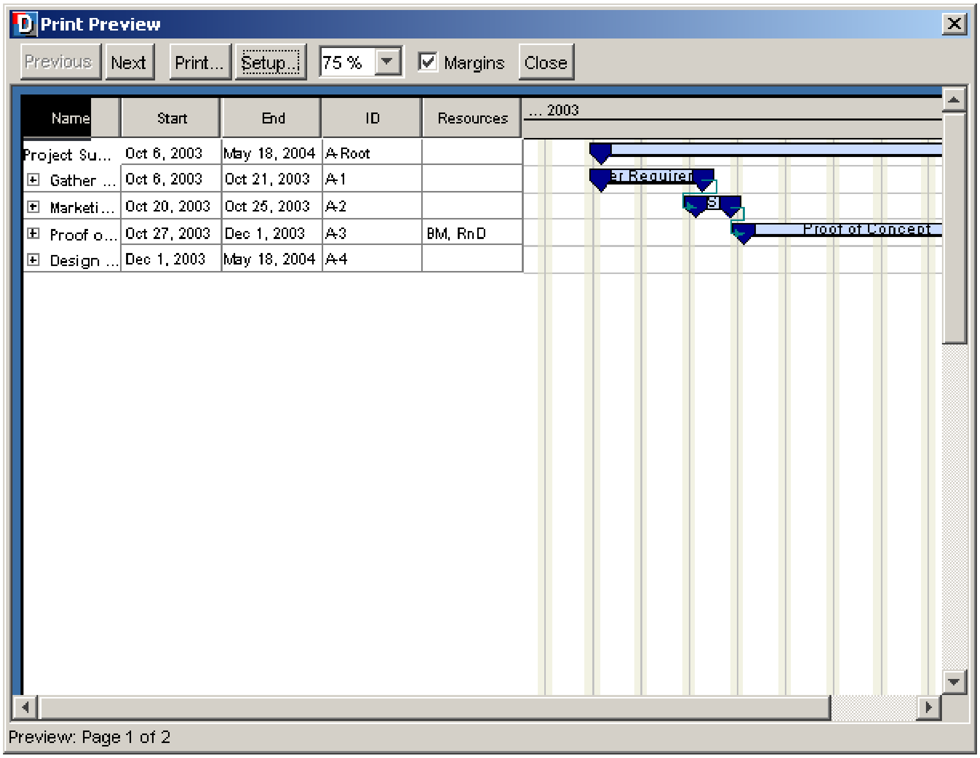
Field	Default Value	Description
Print Range, Dates, From	The chart visible time	The start date of the first printed page. This defines the beginning of the printed data.
Print Range, Dates, To	The chart visible time + the chart visible duration	The end date for the last printed page in a band. This defines the end of the printed data.
Print Range, Fit to	2	The JViews Gantt printing framework provides support for multipage printing through the "pages per band" property. This represents the number of pages you want printed between the start and the end date.
Table, Print **<n>** table column(s)	The chart table column count	Indicates the number of table columns to be printed.
Table, Table size	0.5	Defines the position on the page that separates the table and the Gantt sheet (the value must be between 0 and 1).
Table, Print table on all pages	False	Indicates whether the table should be printed repeatedly on every page.

Table

The table parameters require some further explanation. The Zoom applied to the table is a physical zoom. What is printed, depends on the way the content is displayed on the screen, so you need to prepare the Gantt chart for printing. Take the project `projecttasks.igpr`, which is located in:

```
<installdir>/jviews-gantt86/bin/designer/data/examples
```

Make sure that the five table columns are displayed in Editing Mode. Reduce the margins to **0.0** in the Page Layout of Page Setup, but leave the default values for all the other print settings. The following figure shows the Print Preview. The chart will be printed over two pages.



75 %

☑ Margins

Name	Start	End	ID	Resources
Project Su...	Oct 6, 2003	May 18, 2004	A-Root	
⊞ Gather ...	Oct 6, 2003	Oct 21, 2003	A-1	
⊞ Marketi...	Oct 20, 2003	Oct 25, 2003	A-2	
⊞ Proof o...	Oct 27, 2003	Dec 1, 2003	A-3	BM, RnD
⊞ Design ...	Dec 1, 2003	May 18, 2004	A-4	

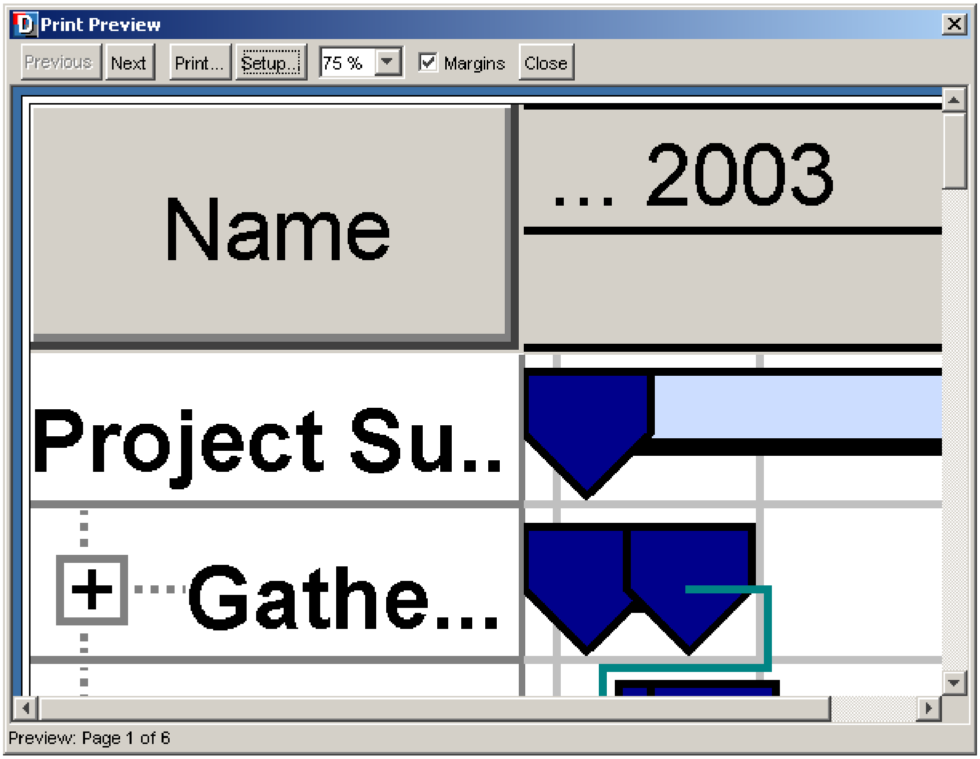
Preview: Page 1 of 2

Experiment by reducing the number of table columns to be printed.

1. Click the **Setup** tab in Print Preview.
2. Click the `Sheet` tab.
3. Set Print **1** table column(s).
4. Leave the other settings at the default values.

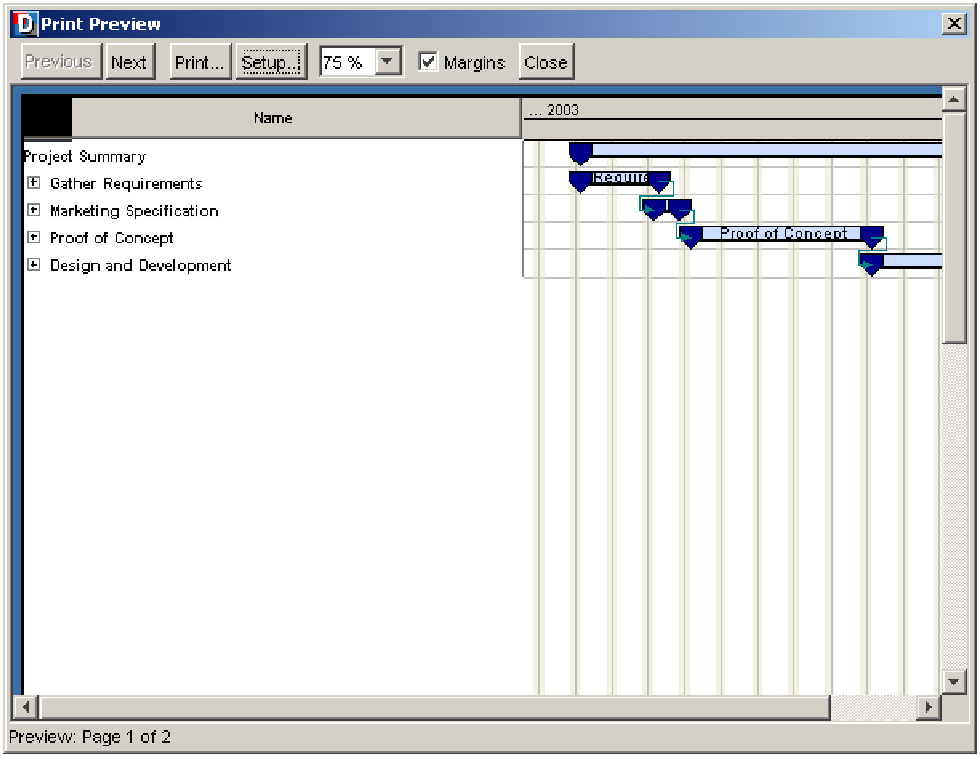
   Note that the table size is set at 50%.
5. Click **OK**.

The following figure shows the Print Preview. Note that the column and the portion of the sheet printed on the page are enlarged several times both widthways and in height, but the column to be printed displays only the content that could be seen in the chart. The ellipses for incomplete words are still visible; the words are not filled in. The chart will be printed over six pages.



Now, adjust the table so that only one column is displayed in the Gantt chart in Editing Mode and you can see the words in full. Leave the print settings as they were in the previous example. Select **Print Preview**.

The following figure shows the Print Preview of a one-column table display after preparation of the display in the chart. The chart will be printed over two pages.

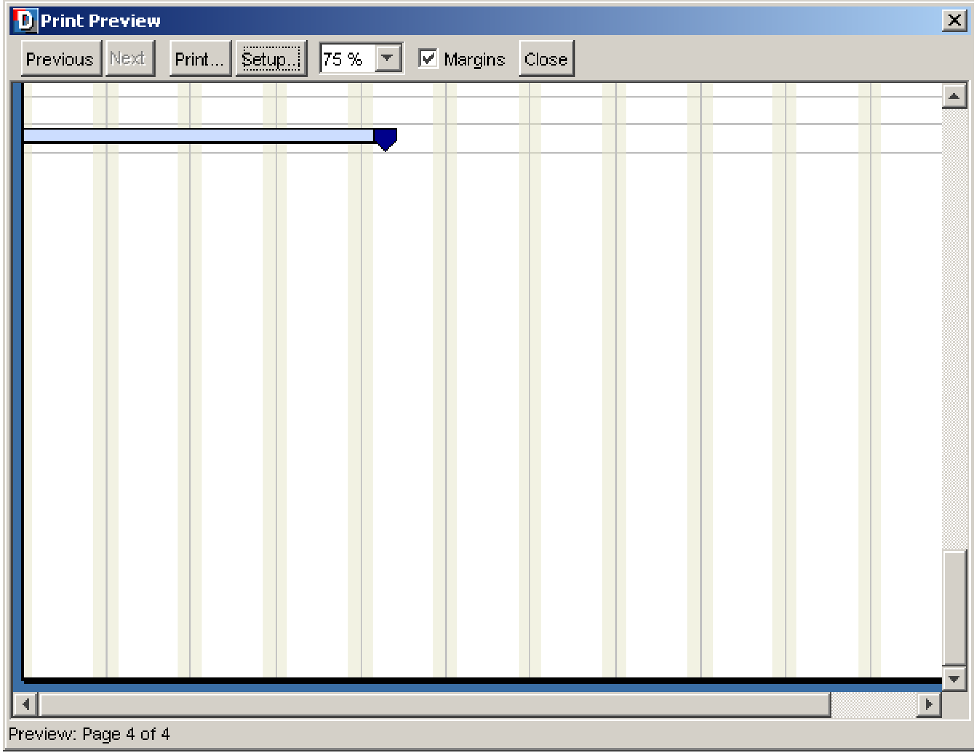


The lesson to be learned is:

- Always preview the results of the print settings before printing.
- Prepare your chart display in Editing Mode before printing.
- Be wary of the effects of the physical zoom on the table.

Print Range

If you now set the Print Range to Fit to: **4** page(s) wide, you obtain more space for time in the sheet. The zoom is intelligent and extends the activity bars to be printed over the four pages. However, the physical zoom on the table does not cause the column or columns to be enlarged. On pages 2 to 4, no table is printed, but only the time scale and the extended activity bars in the Gantt sheet. The following figure shows the Print Preview of the last page, page four, of the example with this setting.



Preview: Page 4 of 4

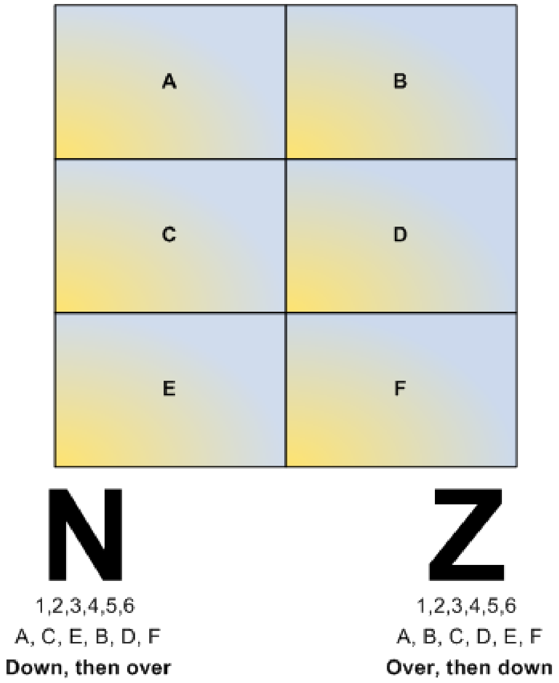
Page order

You can also specify the order in which pages are numbered and printed.

- **Down, then over**: prints top to bottom; the pages are printed in the vertical plane, so that the leftmost pages are printed first.
- **Over, then down**: prints left to right; the pages are printed in the horizontal plane, so that printing spans right across the chart and then proceeds to the next horizontal span of pages below, and so on.

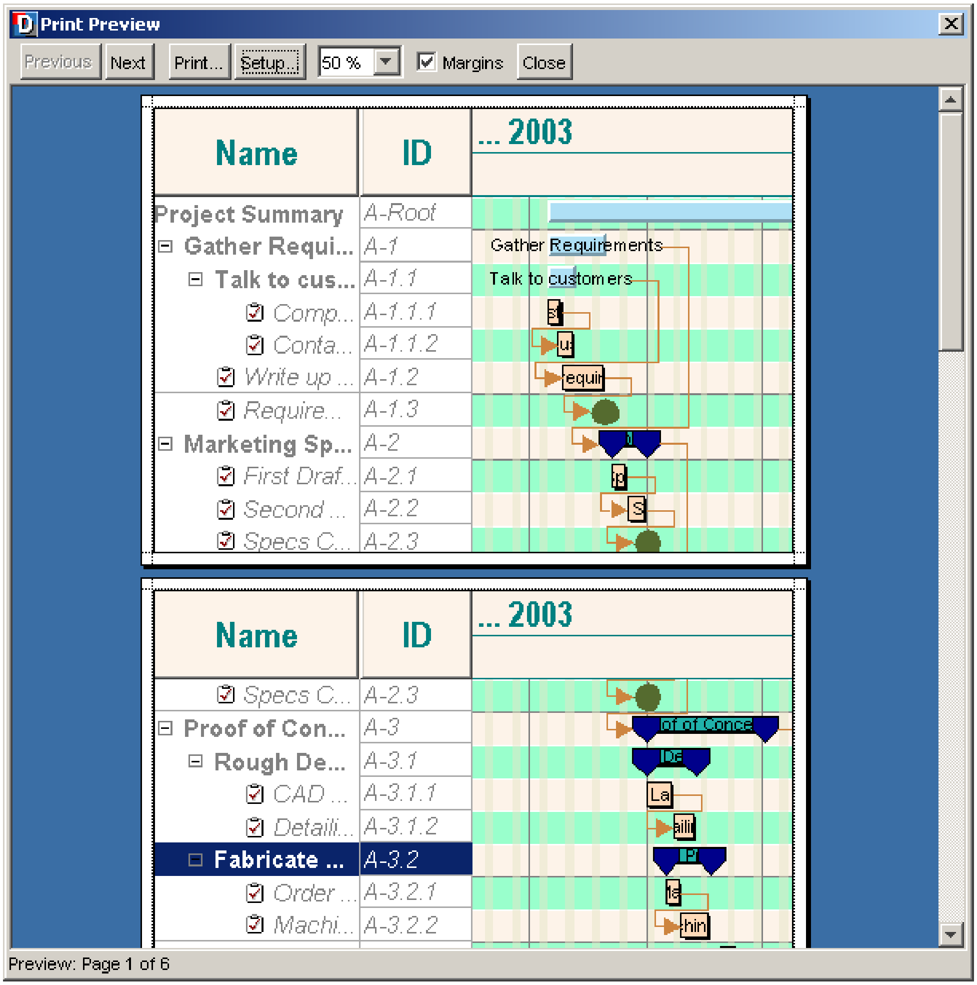
The printing order is different, but the page-numbering is the same, whichever option you select.

The following figure shows the graphical reference for the page-printing order of a six-page document. The sequence of pages is shown as running from A to F, which corresponds to page-numbering from 1 to 6. We will refer the printing order set through Down, then over and Over, then down to this base graphic.



Printing top to bottom

Print the Gantt chart developed in this Getting Started example with the **Down, then over** option set, with the chart printed on six pages. The following figure shows the page-printing order by way of the print preview of the first two pages. You can see that it prints all the way down the leftmost part of the Gantt chart, before working its way across the next horizontal span of the chart. r by way of the print preview of the first two pages. You can see that it prints all the way down the leftmost part of the Gantt chart, before working its way across the next horizontal span of the chart.



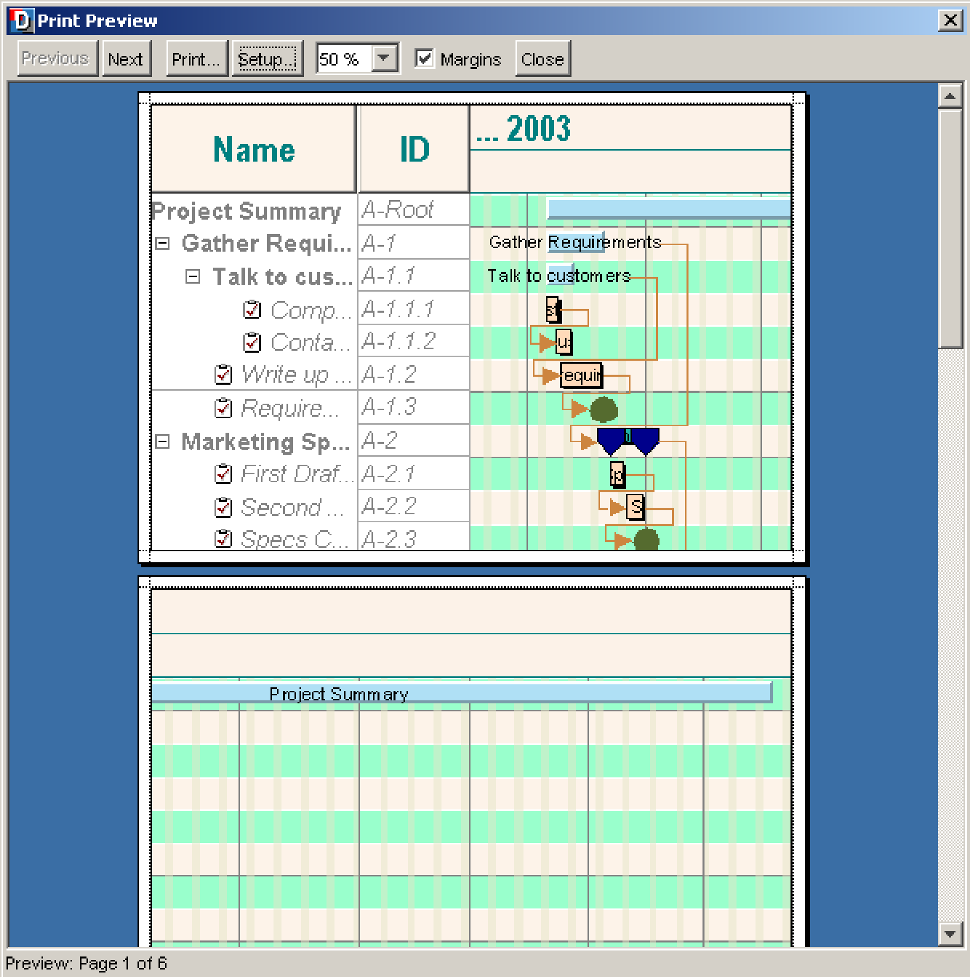
Name	ID
Project Summary	A-Root
⊟ Gather Requi...	A-1
⊟ Talk to cus...	A-1.1
Comp...	A-1.1.1
Conta...	A-1.1.2
Write up ...	A-1.2
Require...	A-1.3
⊟ Marketing Sp...	A-2
First Draf...	A-2.1
Second ...	A-2.2
Specs C...	A-2.3

Name	ID
Specs C...	A-2.3
⊟ Proof of Con...	A-3
⊟ Rough De...	A-3.1
CAD ...	A-3.1.1
Detaili...	A-3.1.2
⊟ Fabricate ...	A-3.2
Order ...	A-3.2.1
Machi...	A-3.2.2

Preview: Page 1 of 6

Printing left to right

Print the Gantt chart developed in this Getting Started example with the **Over, then down** option set, with the chart printed on six pages. The following figure shows the page-printing order by way of the print preview of the first two pages. You can see that it prints all the way across the first part of the Gantt chart in the horizontal plane, before working its way down to the next horizontal span of the chart.



Name	ID	... 2003
Project Summary	A-Root	
⊟ Gather Requi...	A-1	
⊟ Talk to cus...	A-1.1	
☑ Comp...	A-1.1.1	
☑ Conta...	A-1.1.2	
☑ Write up ...	A-1.2	
☑ Require...	A-1.3	
⊟ Marketing Sp...	A-2	
☑ First Draf...	A-2.1	
☑ Second ...	A-2.2	
☑ Specs C...	A-2.3	

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. See *Printing top to bottom* and *Printing left to right* for more information about the Print Preview facility.

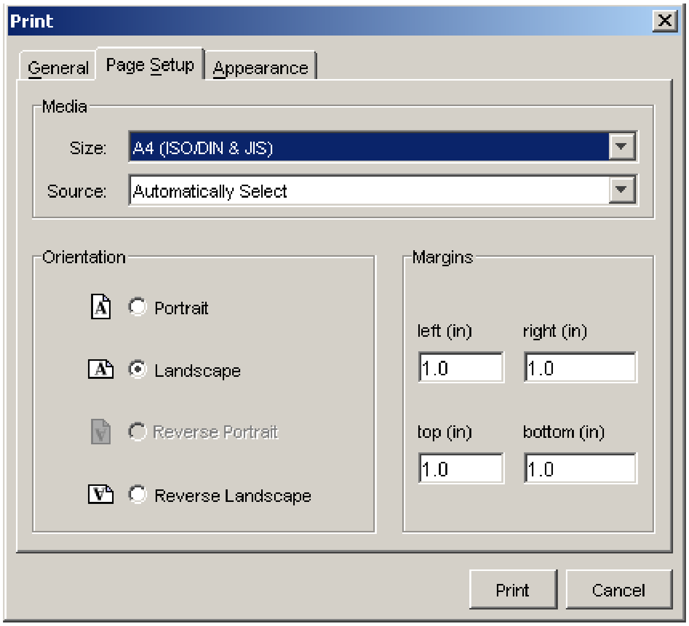
To open the Print Preview window:

- Do one of the following:
  - Click **File > Print Preview**.
  - Click **Print Preview** in the Page Setup window.

Print

You can print one page, all the pages, or selected pages. You can directly control the Page Layout characteristics. You can also specify the printed appearance, by setting features such as color printing and double-sided printing or number of pages per side of paper.

The following figure shows how to set the printing parameters.



Media

Size: A4 (ISO/DIN & JIS)

Source: Automatically Select

Orientation

○ Portrait

● Landscape

○ Reverse Portrait

○ Reverse Landscape

Margins

left (in) 1.0

right (in) 1.0

top (in) 1.0

bottom (in) 1.0

To print a Gantt chart to the default printer from the application:

1. Do one of the following:
   - Click **File > Print**.
   - Type `Ctrl-P`,
   - Open the Print Preview window by **Clicking File > Print Preview**.
   - Open the Page Setup window by **Clicking File > Page Setup**.
2. Click **Print**.

Using more Designer features

Shows you more development features that you can use in the Designer.

In this section

Loading data from a flat file

Describes the tasks necessary to load data form a flat file.

Loading data from a database

Describes the tasks necessary to load data from a database.

Loading user-defined data

Describes the tasks necessary to load data from user-defined data.

Reconfiguring your data

Explains how to change the data and the mapping for an existing chart.

Creating Gantt charts from templates

Explains how to use a predefined project with a style sheet designed for a specific type of application or style of presentation as a starting point for a new Gantt chart.

Using predefined complex renderers

Describes how to customize prebuilt complex renderers made up of composite graphic objects.

Creating a complex renderer

Explains how to create a complex renderer using composite graphics.

Loading data from a flat file

Describes the tasks necessary to load data form a flat file.

In this section

Tasks for data loading

Describes the tasks necessary to load data from a text file with the extension `.csv`.

Accessing the data

Explains how to load data using the New Gantt Chart Wizard.

Mapping the data

Describes how to define the mapping between the properties in the Gantt data model and the columns of the tables in a file.

Tasks for data loading

JViews Gantt lets you load data from a text file with the extension `.csv` (comma separated values). The value separator does not have to be comma (,). The Designer recognizes the following separators: comma, semicolon (;), space, or tab. The use case is based on data from the file `harbor.csv`.

In this section you will:

- Create a chart from basic elements.
- Choose a flat file data source and use `harbor.csv` to provide the data.
- Map the predefined properties recognized in the Gantt data model to columns in the Activities Table.
- Create new properties and map them to the appropriate columns.

The mapped properties can be used for styling your data by creating style rules that use the properties.

Accessing the data

Load the data from the New Gantt Chart Wizard.

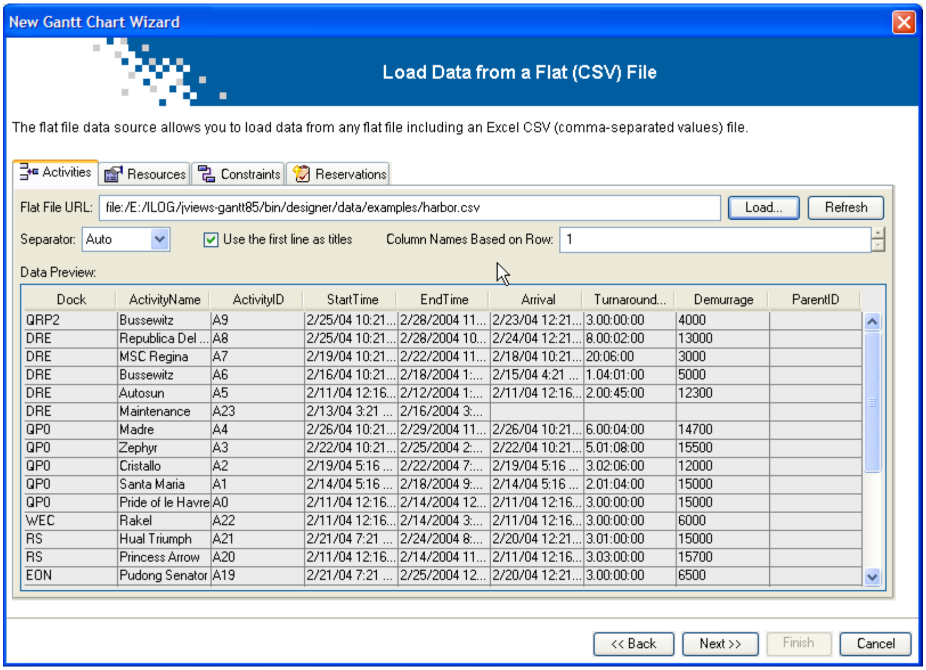
1. Start the New Gantt Chart Wizard, for example, by using the key sequence `Ctrl-w`.
2. Select **Start From Basic Elements** and click **Next**.
3. Select the option **Flat File** by clicking **Next**.
4. Select and open the file `harbor.csv`.

   The data in this file is displayed in the Activities Table.

   Note that the separator value is set to auto. This setting allows the separator used in the flat file to be identified automatically, provided that it is one of the four allowed selectors. The separator used in `harbor.csv` is the semicolon (;). Use of the auto setting means that you do not have to select the semicolon explicitly before loading the file.

5. Click **Next** to access the data-mapping facilities.

   The following figure shows the Activities Table from `harbor.csv`.



The flat file data source allows you to load data from any flat file including an Excel CSV (comma-separated values) file.

Flat File URL: file:/E:/ILOG/jviews-gantt85/bin/designer/data/examples/harbor.csv

Separator: Auto ☑ Use the first line as titles Column Names Based on Row: 1

Data Preview:

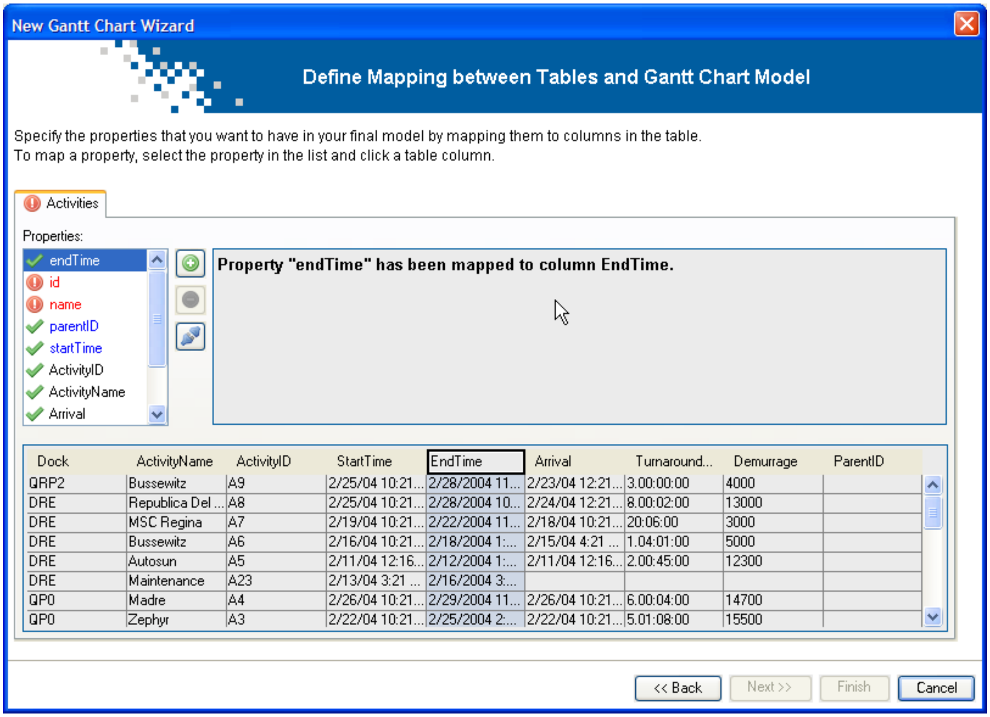
Dock	ActivityName	ActivityID	StartTime	EndTime	Arrival	Turnaround...	Demurrage	ParentID
QRP2	Bussewitz	A9	2/25/04 10:21...	2/28/2004 11...	2/23/04 12:21...	3.00:00:00	4000	
DRE	Republica Del ...	A8	2/25/04 10:21...	2/28/2004 10...	2/24/04 12:21...	8.00:02:00	13000	
DRE	MSC Regina	A7	2/19/04 10:21...	2/22/2004 11...	2/18/04 10:21...	20:06:00	3000	
DRE	Bussewitz	A6	2/16/04 10:21...	2/18/2004 1:...	2/15/04 4:21 ...	1.04:01:00	5000	
DRE	Autosun	A5	2/11/04 12:16...	2/12/2004 1:...	2/11/04 12:16...	2.00:45:00	12300	
DRE	Maintenance	A23	2/13/04 3:21 ...	2/16/2004 3:...				
QP0	Madre	A4	2/26/04 10:21...	2/29/2004 11...	2/26/04 10:21...	6.00:04:00	14700	
QP0	Zephyr	A3	2/22/04 10:21...	2/25/2004 2:...	2/22/04 10:21...	5.01:08:00	15500	
QP0	Cristallo	A2	2/19/04 5:16 ...	2/22/2004 7:...	2/19/04 5:16 ...	3.02:06:00	12000	
QP0	Santa Maria	A1	2/14/04 5:16 ...	2/18/2004 9:...	2/14/04 5:16 ...	2.01:04:00	15000	
QP0	Pride of le Havre	A0	2/11/04 12:16...	2/14/2004 12...	2/11/04 12:16...	3.00:00:00	15000	
WEC	Rakel	A22	2/11/04 12:16...	2/14/2004 3:...	2/11/04 12:16...	3.00:00:00	6000	
RS	Hual Triumph	A21	2/21/04 7:21 ...	2/24/2004 8:...	2/20/04 12:21...	3.01:00:00	15000	
RS	Princess Arrow	A20	2/11/04 12:16...	2/14/2004 11...	2/11/04 12:16...	3.03:00:00	15700	
EON	Pudong Senator	A19	2/21/04 7:21 ...	2/25/2004 12...	2/20/04 12:21...	3.00:00:00	6500	

Mapping the data

You need to define the mapping between the properties in the Gantt data model and the columns of the tables in the file. This mapping is done automatically whenever possible. The table columns from the data source you select are matched against the properties in the Gantt model and if a property with the same name exists they are mapped. If a given property is not found, one with the same name as the column is added and automatically mapped.

In this example, you will map predefined properties to columns in the Activities Table. Properties that are unmapped and mandatory are shown with a red exclamation mark. If required, you can create user-defined properties and map them to other columns in the Activities Table.

The following table shows how to define a mapping.



Define Mapping between Tables and Gantt Chart Model

Specify the properties that you want to have in your final model by mapping them to columns in the table.
To map a property, select the property in the list and click a table column.

Property "endTime" has been mapped to column EndTime.

Dock	ActivityName	ActivityID	StartTime	EndTime	Arrival	Turnaround...	Demurrage	ParentID
QRP2	Bussewitz	A9	2/25/04 10:21...	2/28/2004 11...	2/23/04 12:21...	3.00:00:00	4000	
DRE	Republica Del ...	A8	2/25/04 10:21...	2/28/2004 10...	2/24/04 12:21...	8.00:02:00	13000	
DRE	MSC Regina	A7	2/19/04 10:21...	2/22/2004 11...	2/18/04 10:21...	20:06:00	3000	
DRE	Bussewitz	A6	2/16/04 10:21...	2/18/2004 1:...	2/15/04 4:21 ...	1.04:01:00	5000	
DRE	Autosun	A5	2/11/04 12:16...	2/12/2004 1:...	2/11/04 12:16...	2.00:45:00	12300	
DRE	Maintenance	A23	2/13/04 3:21 ...	2/16/2004 3:...				
QP0	Madre	A4	2/26/04 10:21...	2/29/2004 11...	2/26/04 10:21...	6.00:04:00	14700	
QP0	Zephyr	A3	2/22/04 10:21...	2/25/2004 2:...	2/22/04 10:21...	5.01:08:00	15500	

Mapping the predefined properties

1. To map a property to a column, select the property and click the column or the column header you want to map it to. The name of the property turns blue when it has been mapped.

   An activity is defined by the following properties:

   - `parentID`
   - `id`

- name

- startTime

- endTime

Activities are hierarchical in the Gantt data model. The hierarchy unfolds from the root activity, so that each activity shown in a chart has a parent activity and can in turn be a parent of another activity, unless it is a leaf activity, which has no child activities. Therefore, each activity except the root activity can be associated with the ID of its parent activity.

Each activity is identified by an ID and a name. Its start and end time determine its duration. If these are the same, the activity is a zero-duration activity, which is a milestone.

2. In this example you only need to map the following properties:

Gantt Data Model	Flat File Activities Table
`id`	ActivityID
`name`	ActivityName

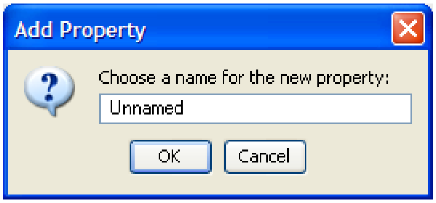
Creating user-defined properties and mapping them

There can be columns in the Activities Table that define additional attributes that do not correspond to any properties in the Gantt data model. You can create user-defined properties for mapping to these columns. In this example, the user-defined properties **Dock**, **Demurrage**, **Arrival**, and **TurnaroundTime** have been automatically created and mapped to the corresponding columns.

If you need to create other user-defined properties to match other columns in the table, add them as follows:

1. Click the large plus sign just to the right of the list of properties ⊕.

   The following figure shows how to create a user-defined property.



Choose a name for the new property:

Unnamed

2. Type the name of the property, for example, **Dock** and click **OK**.

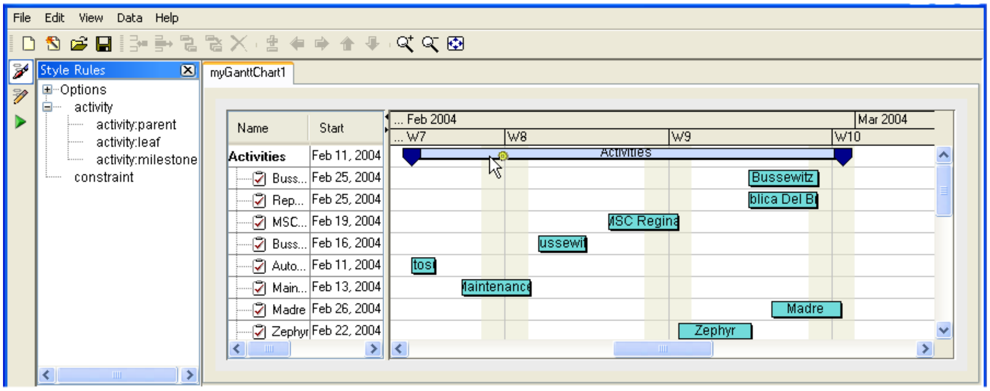
3. Select the newly created property **Dock** and click the column or column header that you want to map it to, in this case, **Dock**.

4. If you make a mistake creating a property, click the large minus sign to remove it ⊖.

5. If you map a property incorrectly, click the clear mapping sign to undo the mapping .

You have mapped all the properties needed to define an activity for the data in `harbor.csv`. Click **Finish** to take a look at the Gantt chart you have created and the data that it contains.

The following figure shows the Gantt chart with activity data from `harbor.csv`.



Name	Start
Activities	Feb 11, 2004
Buss...	Feb 25, 2004
Rep...	Feb 25, 2004
MSC...	Feb 19, 2004
Buss...	Feb 16, 2004
Auto...	Feb 11, 2004
Main...	Feb 13, 2004
Madre	Feb 26, 2004
Zephyr	Feb 22, 2004

There are no constraints mapped for the data in `harbor.csv`, so you can *only* expand the style rules for activity data. You can style the data by using the mapped properties in style rules. See *Create style rule*.

Loading data from a database

Describes the tasks necessary to load data from a database.

In this section

Overview

JViews Gantt lets you connect to a database to load data through JDBC. You can connect, for example, to a Microsoft® Access or an Oracle® database, or to a Microsoft® Excel or a Microsoft® Project file. The use case is based on data from the Microsoft® Excel file `harbor.xls`. The data is the same as in harbor.csv (see the use case in *Loading data from a flat file*), but in this use case you will split the data into three tables with SQL queries.

In this section you will:

- Create a chart from basic elements.
- Choose a JDBC data source.
- Create a new database configuration and use `harbor.xls` to provide the data.
- Perform an SQL query for each data type to fill each table.
- Map the database columns to graphic properties of activities.
- Create new properties and map them to the appropriate columns.

The mapped properties can be used for styling your data by creating style rules that refer to the properties.

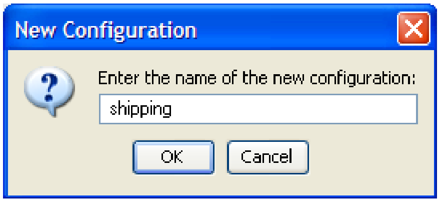
Accessing the data

Load the data from the New Gantt Chart Wizard.

If your data is in a local file, 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.

1. Start the New Gantt Chart Wizard, for example, by choosing **File > New From Wizard**.
2. Select **Start From Basic Elements** and click **Next**.
3. Select **Database (JDBC)** as data source type and click **Next**.
4. Create a new database configuration by clicking the sign. Type the name **shipping** and click **OK**.

   The following figure shows how to name a new configuration.

   New Configuration

   Enter the name of the new configuration: shipping

5. For the database URL, click the Ellipsis button and select the Microsoft® Excel version of the harbor example, `harbor.xls`.

   This operation provokes an error because you need to formulate an appropriate SQL query for the file type and content.

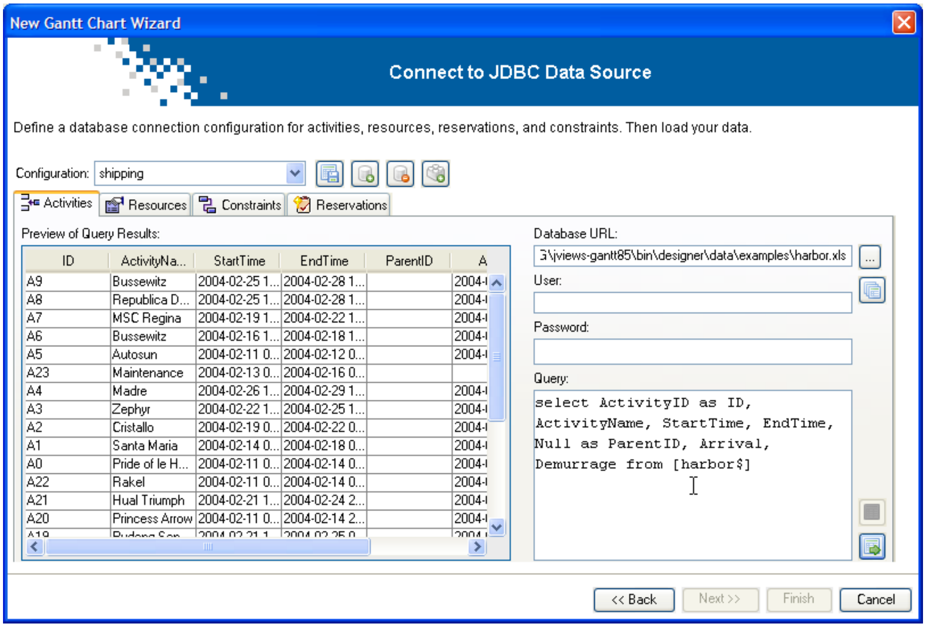
6. Click the Copy sign to reuse the same database URL for the other tables, since all the data is in the same file.
7. With the **Activities Table** tab active, formulate an SQL query by typing:

   ```
   select ActivityID as ID, ActivityName, StartTime, EndTime, Null as
   ParentID,
   Arrival, Demurrage from [harbor$]
   ```

   This query names the table columns in the Designer and determines their content. For example, the content of the ActivityID column in the data file is written to ID. The column headed ParentID is set to null and therefore appears empty; this is necessary because there is no data for ParentID in the data file. The remaining columns are named as in the data file. This data is taken from the Microsoft® Excel page named `harbor`.

8. Click the execute query sign to execute the query .

   The following figure shows the result of the activity query.



New Gantt Chart Wizard

Connect to JDBC Data Source

Define a database connection configuration for activities, resources, reservations, and constraints. Then load your data.

Configuration: shipping

Preview of Query Results:

ID	ActivityNa...	StartTime	EndTime	ParentID	A
A9	Bussewitz	2004-02-25 1...	2004-02-28 1...		2004-(
A8	Republica D...	2004-02-25 1...	2004-02-28 1...		2004-(
A7	MSC Regina	2004-02-19 1...	2004-02-22 1...		2004-(
A6	Bussewitz	2004-02-16 1...	2004-02-18 1...		2004-(
A5	Autosun	2004-02-11 0...	2004-02-12 0...		2004-(
A23	Maintenance	2004-02-13 0...	2004-02-16 0...		
A4	Madre	2004-02-26 1...	2004-02-29 1...		2004-(
A3	Zephyr	2004-02-22 1...	2004-02-25 1...		2004-(
A2	Cristallo	2004-02-19 0...	2004-02-22 0...		2004-(
A1	Santa Maria	2004-02-14 0...	2004-02-18 0...		2004-(
A0	Pride of le H...	2004-02-11 0...	2004-02-14 0...		2004-(
A22	Rakel	2004-02-11 0...	2004-02-14 0...		2004-(
A21	Hual Triumph	2004-02-21 1...	2004-02-24 2...		2004-(
A20	Princess Arrow	2004-02-11 0...	2004-02-14 2...		2004-(
A19	Pudong Sen...	2004-02-21 1...	2004-02-25 0...		2004-(

Database URL: 3\jviews-gantt85\bin\designer\data\examples\harbor.xls

User:

Password:

Query:

```
select ActivityID as ID,
ActivityName, StartTime, EndTime,
Null as ParentID, Arrival,
Demurrage from [harbor$]
```

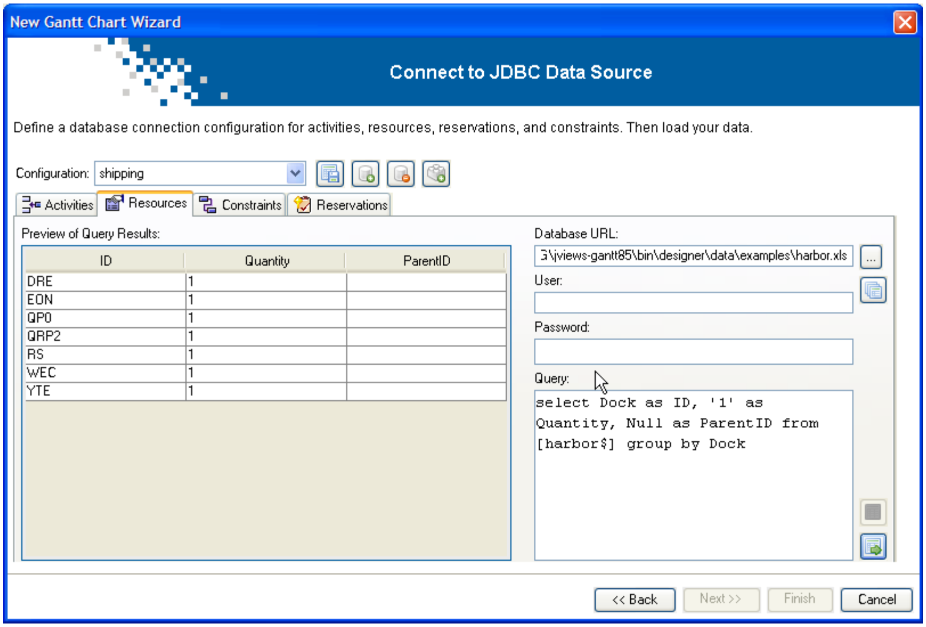
9. Select the **Resources Table** tab and formulate the query:

```
select Dock as ID, '1' as Quantity, Null as ParentID from [harbor$] group
by Dock
```

10. Click the execute query sign to execute the query .

In this use case, unlike the previous one, the dock is represented as a resource. The resource data is aggregated (`group by` clause) so that each dock only appears once.

The following figure shows the result of the resource query.



New Gantt Chart Wizard

Connect to JDBC Data Source

Define a database connection configuration for activities, resources, reservations, and constraints. Then load your data.

Configuration: shipping

Preview of Query Results:

ID	Quantity	ParentID
DRE	1	
EON	1	
QPO	1	
QRP2	1	
RS	1	
WEC	1	
YTE	1	

Database URL: 3\jviews-gantt85\bin\designer\data\examples\harbor.xls

User:

Password:

Query:

```
select Dock as ID, '1' as
Quantity, Null as ParentID from
[harbor$] group by Dock
```

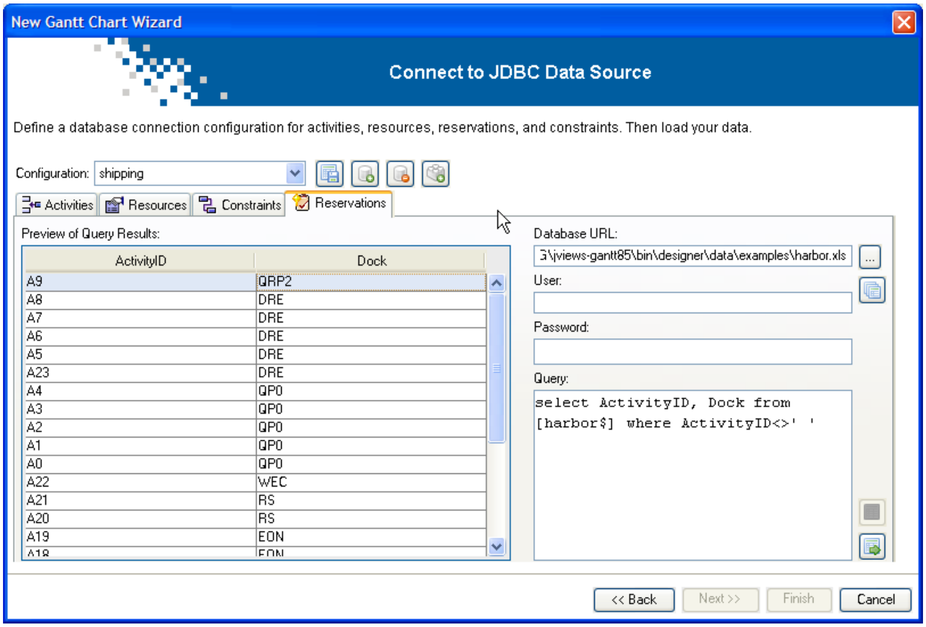
11. Select the **Reservations Table** tab and formulate the query:

```
select ActivityID, Dock from [harbor$] where ActivityID<>' '
```

12. Click the execute query sign to execute the query.

    The data loaded excludes any rows where the Activity ID is blank, indicating the dock is not in use (`where` clause).

    The following figure shows the result of the reservation query.



Connect to JDBC Data Source

Define a database connection configuration for activities, resources, reservations, and constraints. Then load your data.

Configuration: shipping

Preview of Query Results:

ActivityID	Dock
A9	QRP2
A8	DRE
A7	DRE
A6	DRE
A5	DRE
A23	DRE
A4	QPO
A3	QPO
A2	QPO
A1	QPO
A0	QPO
A22	WEC
A21	RS
A20	RS
A19	EON
A18	EON

Database URL: 3\jviews-gantt85\bin\designer\data\examples\harbor.xls

User:

Password:

Query:

```
select ActivityID, Dock from
[harbor$] where ActivityID<>' '
```

13. Select the **Constraints Table** tab and delete the query because you do not want any constraints loaded.

14. Click the execute query sign to execute the empty query .

15. Click **Next** to access the data-mapping facilities.

Mapping the data

You need to define the mapping between the properties in the Gantt data model and the columns of the tables. Properties that are unmapped and mandatory are shown with a red exclamation mark.

Mapping properties to the Activities Table

See in *Mapping the predefined properties* for how to map the predefined property **name** to the corresponding column. Note that in this case `id` is mapped to `ID`.

In this example, the columns **Arrival** and **Demurrage** in the Activities Table define additional attributes that do not correspond to any properties in the Gantt data model, however, the user-defined properties **Arrival**, and **Demurrage** have been automatically created and mapped to these columns.

- Select the **name** property and click the **ActivityName** column or column header to create the mapping.

The properties you have mapped allow you to display the following in a Gantt chart:

- The tasks to be carried out on specific ships
- The duration of these tasks
- The arrival time of each ship
- The cost of mooring the ship

Mapping properties to the Resources Table

In the Resources Table, you only need to map the **name** property. The other properties required have been automatically created and mapped to the corresponding columns.

- Select the **name** property and then click the **ID** column or column header to create the mapping.

The resources are the docks where ships will be moored. The capacity of each dock is just one activity (`quantity` is always `1`).

Mapping properties to the Reservations Table

In the Reservations Table, you only need to map the **resourceID** property. The other properties required have been automatically created and mapped to the corresponding columns.

1. Select the **resourceID** property and then click the **Dock** column or column header to create the mapping.

   Reservations link an activity to a resource. An activity is a task, which in this example consists of performing an operation such as loading, unloading, or repairing a named ship during a specified time. This task is carried out at a specified dock resource.

2. Click the Save Configuration sign to save the database configuration for reuse between projects.

   If you want to reuse your data connection configuration in projects other than the current one, use the Save Configuration icon.

   If the configuration is for use in the current project only, you do not need to use this icon to save it. It is automatically saved when you save the project.

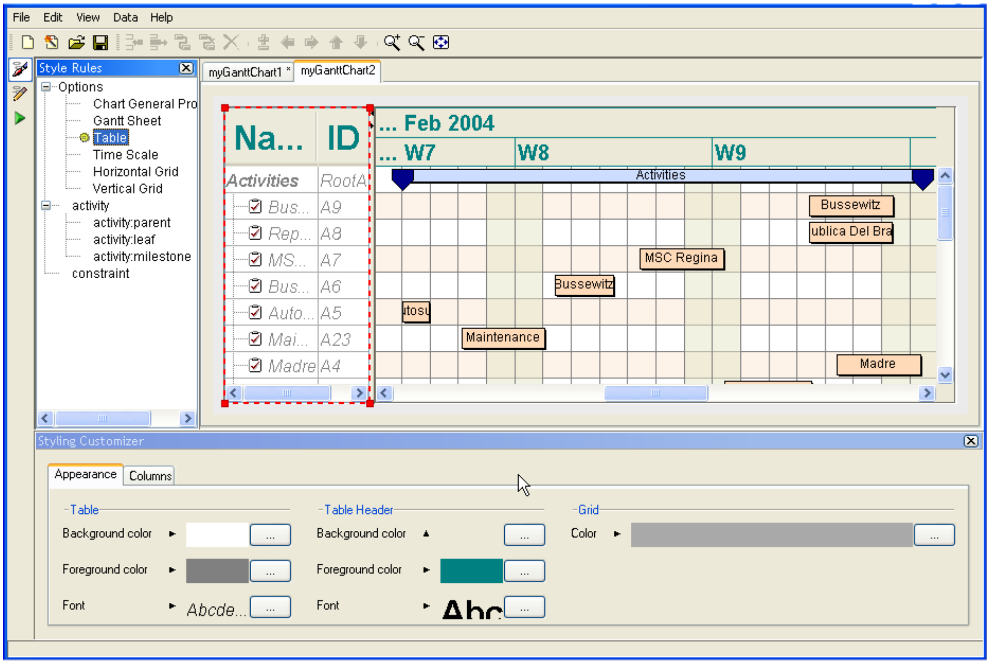
   The IBM® ILOG® predefined connections cannot be saved by clicking the icon. In the case of a predefined connection, the Save Configuration icon is disabled.

3. Click **Next** to choose the type of chart and then to choose a theme.

4. Click **Finish** to view the data in a basic Gantt chart, shown in the following figure.

To save this chart, choose **File > Save As** and enter **harbor1** as the name of the project file.

The following table shows an activity-oriented chart showing the harbor data.



Loading user-defined data

Describes the tasks necessary to load data from user-defined data.

In this section

Overview

Explains the steps necessary to load data from user-defined data.

Accessing the data

Explains how to load data from the New Gantt Chart Wizard.

Mapping the data

Describes how to map the properties of activities in the Gantt user-defined data model to the properties in the Activities Table of the data source.

Overview

It can be useful to copy data from a data source, for example, a Microsoft® Excel file, and paste it into the tables in memory for quick prototyping of a Gantt chart. If you choose this route, you do not need to understand SQL queries or database connections; you just need the data to be in some kind of row-and-column format with separators between the items in a row and a carriage return character at the end of the row. You can then map the properties of the Gantt user-defined data model to the columns in the normal way.

The use case is based on data from the Microsoft® Excel file `harbor.xls`.

In this section you will:

- Create a chart from basic elements.
- Choose a user-defined data source.
- Paste the data from `harbor.xls` into the Activities Table.
- Add a column for the parent ID to make it possible to structure the model in a hierarchy and to map the data.
- Map the columns in the data source to graphic properties of activities.
- Create new properties and map them to the appropriate columns.

The mapped properties can be used for styling your data by creating style rules that refer to the properties.

Accessing the data

Load the data from the New Gantt Chart Wizard.

1. In Microsoft® Excel, open the file `harbor.xls` that contains the data you want to load into memory.

   This file is located in:

   `<installdir>/jviews-gantt86/bin/designer/data/examples`

2. Start the New Gantt Chart Wizard, for example, by choosing **File > New From Wizard**.

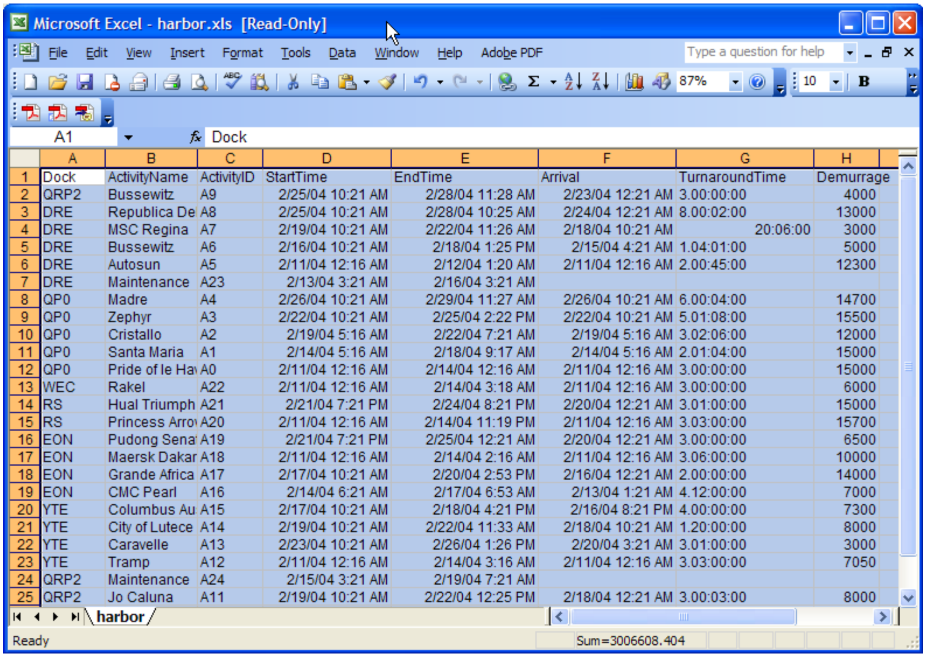
3. Select **Start From Basic Elements** and click **Next**.

4. Select **User-Defined Data Model** as data source type and click **Next**.

   The properties of activities in the default data model are displayed. The default data model also includes properties for resources, constraints, and reservations. Depending on the data you want to load into memory, you need to adapt one or more of these sets of properties. Click the corresponding tab to access the properties of each object in the default data model.

5. Select and copy all the filled rows and columns in the file `harbor.xls`.

   The following figure shows the content of `harbor.xls` being copied.

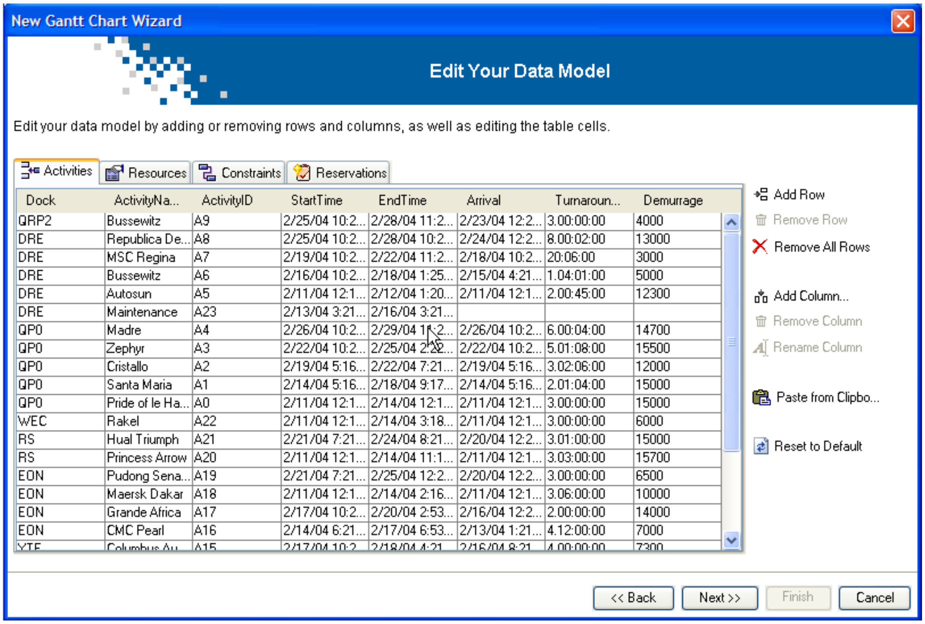


	A	B	C	D	E	F	G	H
1	Dock	ActivityName	ActivityID	StartTime	EndTime	Arrival	TurnaroundTime	Demurrage
2	QRP2	Bussewitz	A9	2/25/04 10:21 AM	2/28/04 11:28 AM	2/23/04 12:21 AM	3.00:00:00	4000
3	DRE	Republica De	A8	2/25/04 10:21 AM	2/28/04 10:25 AM	2/24/04 12:21 AM	8.00:02:00	13000
4	DRE	MSC Regina	A7	2/19/04 10:21 AM	2/22/04 11:26 AM	2/18/04 10:21 AM	20:06:00	3000
5	DRE	Bussewitz	A6	2/16/04 10:21 AM	2/18/04 1:25 PM	2/15/04 4:21 AM	1.04:01:00	5000
6	DRE	Autosun	A5	2/11/04 12:16 AM	2/12/04 1:20 AM	2/11/04 12:16 AM	2.00:45:00	12300
7	DRE	Maintenance	A23	2/13/04 3:21 AM	2/16/04 3:21 AM			
8	QP0	Madre	A4	2/26/04 10:21 AM	2/29/04 11:27 AM	2/26/04 10:21 AM	6.00:04:00	14700
9	QP0	Zephyr	A3	2/22/04 10:21 AM	2/25/04 2:22 PM	2/22/04 10:21 AM	5.01:08:00	15500
10	QP0	Cristallo	A2	2/19/04 5:16 AM	2/22/04 7:21 AM	2/19/04 5:16 AM	3.02:06:00	12000
11	QP0	Santa Maria	A1	2/14/04 5:16 AM	2/18/04 9:17 AM	2/14/04 5:16 AM	2.01:04:00	15000
12	QP0	Pride of le Ha	A0	2/11/04 12:16 AM	2/14/04 12:16 AM	2/11/04 12:16 AM	3.00:00:00	15000
13	WEC	Rakel	A22	2/11/04 12:16 AM	2/14/04 3:18 AM	2/11/04 12:16 AM	3.00:00:00	6000
14	RS	Hual Triumph	A21	2/21/04 7:21 PM	2/24/04 8:21 PM	2/20/04 12:21 AM	3.01:00:00	15000
15	RS	Princess Arro	A20	2/11/04 12:16 AM	2/14/04 11:19 PM	2/11/04 12:16 AM	3.03:00:00	15700
16	EON	Pudong Sena	A19	2/21/04 7:21 PM	2/25/04 12:21 AM	2/20/04 12:21 AM	3.00:00:00	6500
17	EON	Maersk Dakar	A18	2/11/04 12:16 AM	2/14/04 2:16 AM	2/11/04 12:16 AM	3.06:00:00	10000
18	EON	Grande Africa	A17	2/17/04 10:21 AM	2/20/04 2:53 PM	2/16/04 12:21 AM	2.00:00:00	14000
19	EON	CMC Pearl	A16	2/14/04 6:21 AM	2/17/04 6:53 AM	2/13/04 1:21 AM	4.12:00:00	7000
20	YTE	Columbus Au	A15	2/17/04 10:21 AM	2/18/04 4:21 PM	2/16/04 8:21 PM	4.00:00:00	7300
21	YTE	City of Lutece	A14	2/19/04 10:21 AM	2/22/04 11:33 AM	2/18/04 10:21 AM	1.20:00:00	8000
22	YTE	Caravelle	A13	2/23/04 10:21 AM	2/26/04 1:26 PM	2/20/04 3:21 AM	3.01:00:00	3000
23	YTE	Tramp	A12	2/11/04 12:16 AM	2/14/04 3:16 AM	2/11/04 12:16 AM	3.03:00:00	7050
24	QRP2	Maintenance	A24	2/15/04 3:21 AM	2/19/04 7:21 AM			
25	QRP2	Jo Caluna	A11	2/19/04 10:21 AM	2/22/04 12:25 PM	2/18/04 12:21 AM	3.00:03:00	8000

6. Make sure that the Activities Table is selected and click **Paste from Clipboard** to paste the content of the file `harbor.xls` into the Activities Table.

7. Click **OK** to confirm that you want to keep the first row of the copied data as the column headers.

   The following figure shows the resulting data model for the data in `harbor.xls`.



Edit your data model by adding or removing rows and columns, as well as editing the table cells.

Dock	ActivityNa...	ActivityID	StartTime	EndTime	Arrival	Turnaroun...	Demurrage
QRP2	Bussewitz	A9	2/25/04 10:2...	2/28/04 11:2...	2/23/04 12:2...	3.00:00:00	4000
DRE	Republica De...	A8	2/25/04 10:2...	2/28/04 10:2...	2/24/04 12:2...	8.00:02:00	13000
DRE	MSC Regina	A7	2/19/04 10:2...	2/22/04 11:2...	2/18/04 10:2...	20:06:00	3000
DRE	Bussewitz	A6	2/16/04 10:2...	2/18/04 1:25...	2/15/04 4:21...	1.04:01:00	5000
DRE	Autosun	A5	2/11/04 12:1...	2/12/04 1:20...	2/11/04 12:1...	2.00:45:00	12300
DRE	Maintenance	A23	2/13/04 3:21...	2/16/04 3:21...			
QP0	Madre	A4	2/26/04 10:2...	2/29/04 11:2...	2/26/04 10:2...	6.00:04:00	14700
QP0	Zephyr	A3	2/22/04 10:2...	2/25/04 2:22...	2/22/04 10:2...	5.01:08:00	15500
QP0	Cristallo	A2	2/19/04 5:16...	2/22/04 7:21...	2/19/04 5:16...	3.02:06:00	12000
QP0	Santa Maria	A1	2/14/04 5:16...	2/18/04 9:17...	2/14/04 5:16...	2.01:04:00	15000
QP0	Pride of le Ha...	A0	2/11/04 12:1...	2/14/04 12:1...	2/11/04 12:1...	3.00:00:00	15000
WEC	Rakel	A22	2/11/04 12:1...	2/14/04 3:18...	2/11/04 12:1...	3.00:00:00	6000
RS	Hual Triumph	A21	2/21/04 7:21...	2/24/04 8:21...	2/20/04 12:2...	3.01:00:00	15000
RS	Princess Arrow	A20	2/11/04 12:1...	2/14/04 11:1...	2/11/04 12:1...	3.03:00:00	15700
EON	Pudong Sena...	A19	2/21/04 7:21...	2/25/04 12:2...	2/20/04 12:2...	3.00:00:00	6500
EON	Maersk Dakar	A18	2/11/04 12:1...	2/14/04 2:16...	2/11/04 12:1...	3.06:00:00	10000
EON	Grande Africa	A17	2/17/04 10:2...	2/20/04 2:53...	2/16/04 12:2...	2.00:00:00	14000
EON	CMC Pearl	A16	2/14/04 6:21...	2/17/04 6:53...	2/13/04 1:21...	4.12:00:00	7000
YTE	Columbus Au...	A15	2/17/04 10:2...	2/18/04 4:21...	2/16/04 8:21...	4.00:00:00	7300

8. Click **Add Column** type the name **ParentID**, and click **OK**.

   Activities in the Gantt data model are structured in a hierarchy of parent nodes and leaf nodes. The `parentID` property also groups sibling activities under the same parent. In this example, there is no single root activity defined in the tabular data that groups all of the other activities. Therefore, you can leave the `parentID` column empty.

9. Remove all the rows from the Resources, Constraints, and Reservations tables.

   You are only going to test your Gantt chart with activity data, and you should therefore remove the rows from the other tables. If you do not do this, the sample data in these tables will conflict with the activity data that you copied from the Microsoft® Excel worksheet. These data conflicts will prevent you from proceeding with the mapping of the data model properties.

   For example, click the **Resources** tab and then click **Remove All Rows**. Repeat for all the rows in each of the Resources, Constraints, and Reservations tables.

10. Click **Next** to map the data.

Mapping the data

You need to map the properties of activities in the Gantt user-defined data model to the properties in the Activities Table of the data source. Properties that are unmapped and mandatory are shown with a red exclamation mark.

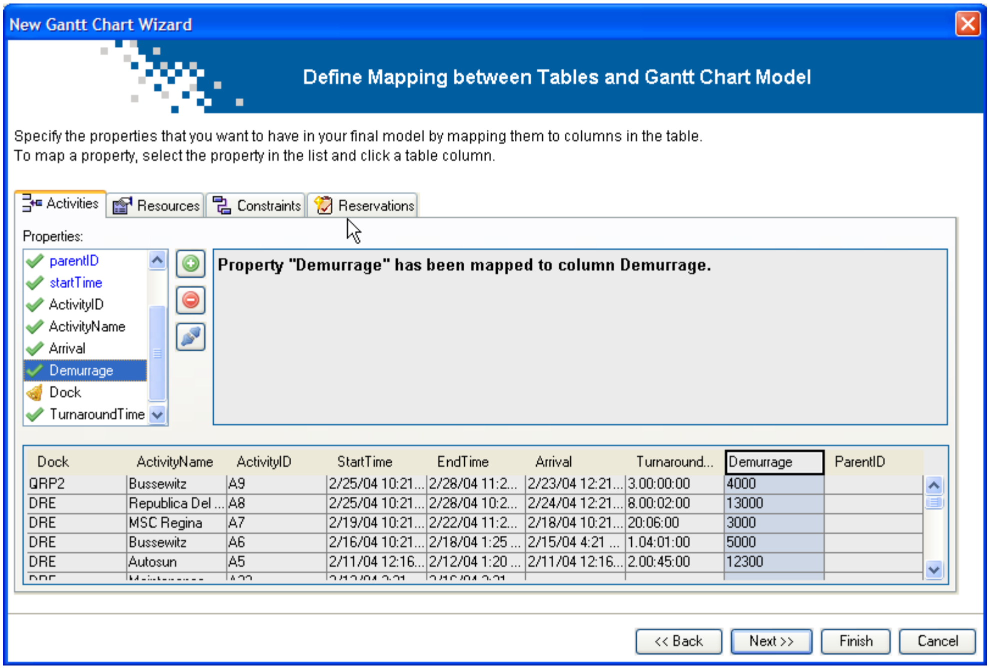
See *Mapping the predefined properties* for how to map the predefined properties **id** and **name** to the corresponding columns in the Activities Table.

The user-defined properties **Arrival**, **TurnaroundTime**, and **Demurrage** have been automatically created and mapped to the corresponding columns in the Activities Table.

Do not map to the column headed **Dock**. The content of this column refers to resources. The objective is to test your Gantt chart with activity data, so you do not need this column.

♦ Select the **Dock** property and click the clear mapping sign to remove the mapping.

The following figure shows the result of the data mapping.



Specify the properties that you want to have in your final model by mapping them to columns in the table.
To map a property, select the property in the list and click a table column.

Property "Demurrage" has been mapped to column Demurrage.

Dock	ActivityName	ActivityID	StartTime	EndTime	Arrival	Turnaround...	Demurrage	ParentID
QRP2	Bussewitz	A9	2/25/04 10:21...	2/28/04 11:2...	2/23/04 12:21...	3.00:00:00	4000	
DRE	Republica Del ...	A8	2/25/04 10:21...	2/28/04 10:2...	2/24/04 12:21...	8.00:02:00	13000	
DRE	MSC Regina	A7	2/19/04 10:21...	2/22/04 11:2...	2/18/04 10:21...	20:06:00	3000	
DRE	Bussewitz	A6	2/16/04 10:21...	2/18/04 1:25 ...	2/15/04 4:21 ...	1.04:01:00	5000	
DRE	Autosun	A5	2/11/04 12:16...	2/12/04 1:20 ...	2/11/04 12:16...	2.00:45:00	12300	

Click **Next** to choose the activity-oriented Gantt chart and then choose a theme. Click **Finish** to view the user-defined data in a basic Gantt chart.

Reconfiguring your data

Explains how to change the data and the mapping for an existing chart.

In this section

Changing data aspects

The data for a Gantt chart is set up when you load and map data in the New Gantt Chart Wizard or open a template. The Designer lets you change the data loaded and the mapping for an existing chart.

If you have an existing chart and you want to include further data from the same data source, or try out a new data set, you can reconfigure the data associated with the chart.

You can change the following aspects of the data associated with the chart:

- The data file
- The data mapping

  If you have changed the data file, you must define a new mapping. If you have the same data file, you can change the mapping anyway, for example, to load extra data.

This section reconfigures the data for the `harbor1` activity-oriented chart; see *Loading data from a database*. It assumes that you have decided to make a new version of the chart that includes the dock and turnaround time, with the activities for the DRE dock displayed showing this time.

First choose **File > Open** and browse to select the `harbor1.igpr` project file.

Choose **Data > Reconfigure Data**. The Reconfigure Data Wizard opens at the Gantt Connect to JDBC Data Source page.

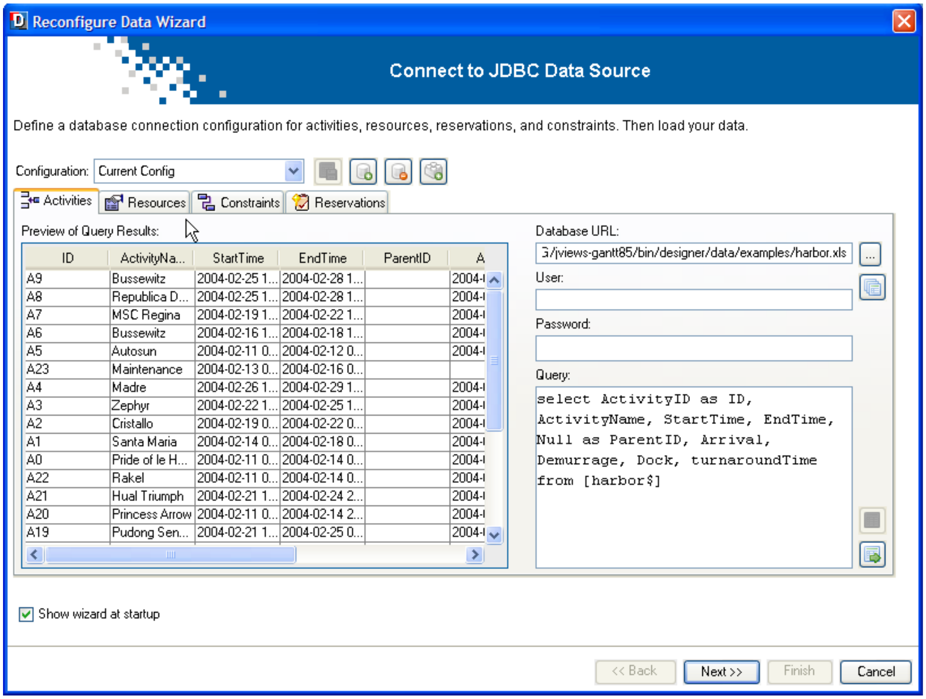
Data connection configuration

The displayed information reflects the current configuration.

To change the SQL query for activities to retrieve extra data:

1. Add `Dock` and `TurnaroundTime` to the columns to retrieve.
2. Click **Execute Query** to load the data and see the new column in the Preview.

   The following figure shows the reconfiguring data: additional columns.



Define a database connection configuration for activities, resources, reservations, and constraints. Then load your data.

Configuration: Current Config

Preview of Query Results:

ID	ActivityNa...	StartTime	EndTime	ParentID	A
A9	Bussewitz	2004-02-25 1...	2004-02-28 1...		2004-
A8	Republica D...	2004-02-25 1...	2004-02-28 1...		2004-
A7	MSC Regina	2004-02-19 1...	2004-02-22 1...		2004-
A6	Bussewitz	2004-02-16 1...	2004-02-18 1...		2004-
A5	Autosun	2004-02-11 0...	2004-02-12 0...		2004-
A23	Maintenance	2004-02-13 0...	2004-02-16 0...		
A4	Madre	2004-02-26 1...	2004-02-29 1...		2004-
A3	Zephyr	2004-02-22 1...	2004-02-25 1...		2004-
A2	Cristallo	2004-02-19 0...	2004-02-22 0...		2004-
A1	Santa Maria	2004-02-14 0...	2004-02-18 0...		2004-
A0	Pride of le H...	2004-02-11 0...	2004-02-14 0...		2004-
A22	Rakel	2004-02-11 0...	2004-02-14 0...		2004-
A21	Hual Triumph	2004-02-21 1...	2004-02-24 2...		2004-
A20	Princess Arrow	2004-02-11 0...	2004-02-14 2...		2004-
A19	Pudong Sen...	2004-02-21 1...	2004-02-25 0...		2004-

Database URL: 3/jviews-gantt85/bin/designer/data/examples/harbor.xls

User:

Password:

Query:

```
select ActivityID as ID,
ActivityName, StartTime, EndTime,
Null as ParentID, Arrival,
Demurrage, Dock, turnaroundTime
from [harbor$]
```

☑ Show wizard at startup

3. Click **Next**.

Data mapping

To make use of the extra data in your diagram, you need to map it. In this example, the properties **Dock** and **TurnaroundTime** have been automatically created and mapped to the **Dock** and **TurnaroundTime** columns.

To exit the wizard:

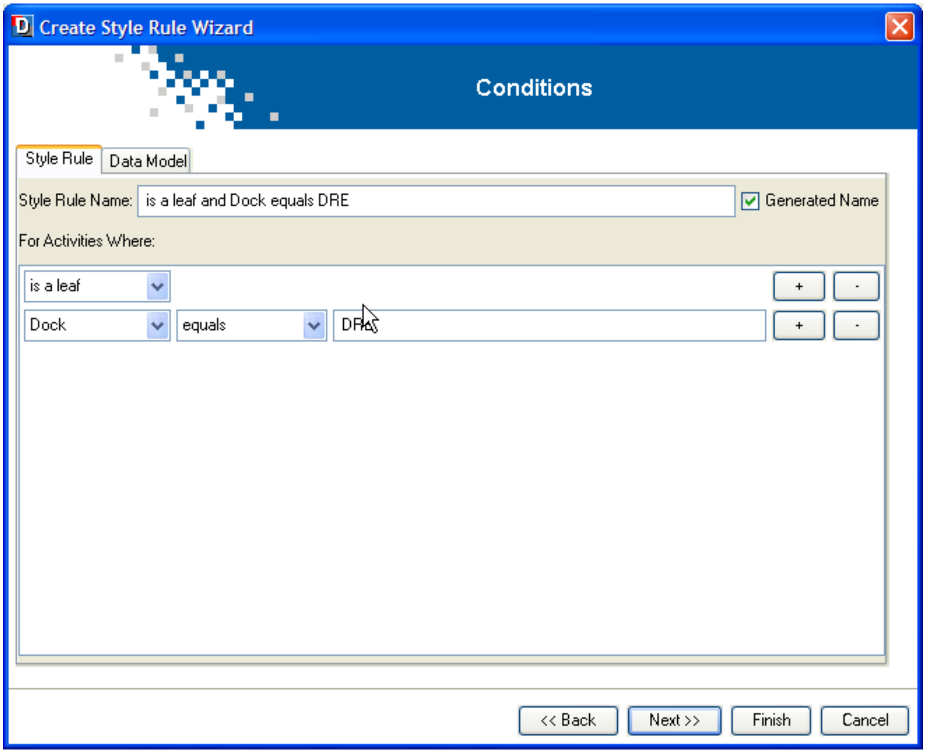
- Click **Finish**.

Styling the DRE dock activities

To style the DRE dock activities:

1. Select the `activity:leaf` rule.
2. Click **Edit > Create Style Rule**.
3. Create a rule on the DRE dock.

   The following figure shows a style rule on the additional attribute.



Create Style Rule Wizard

Conditions

Style Rule Name: is a leaf and Dock equals DRE ☑ Generated Name

For Activities Where:

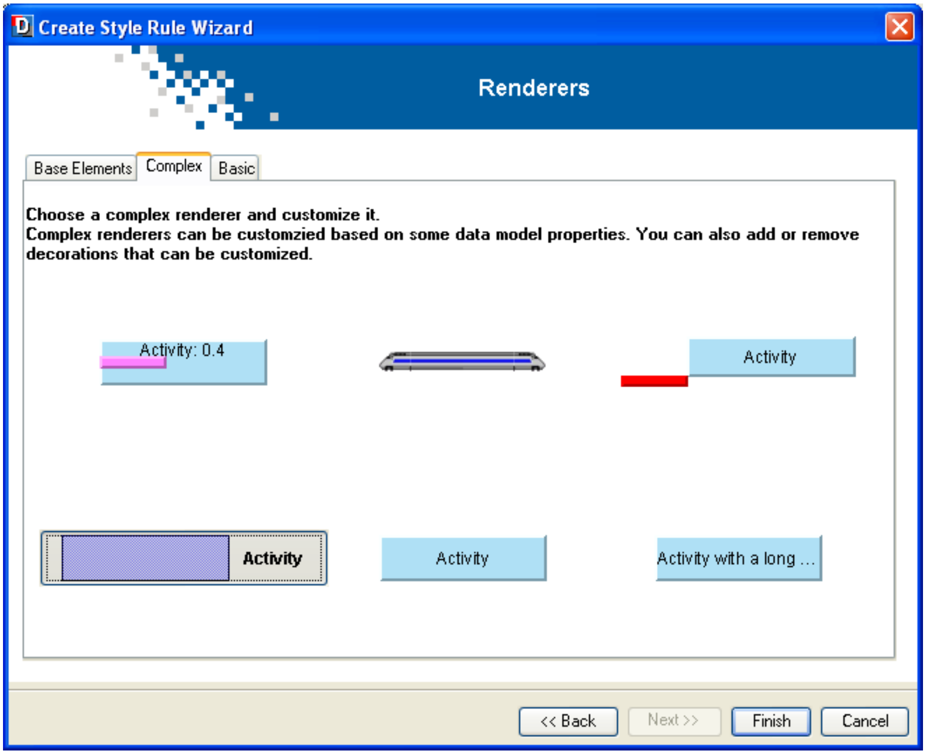
is a leaf

Dock equals DRE

4. Click **Next** and select the **New renderer** option.
5. Click **Next** .
6. In the Renderers page, **Complex** tab, select the activity bar with SVG and text.

   This complex renderer will allow you to select a different vector image (SVG format) and to customize the associated text.

   The following figure shows the selection of the complex renderer for an activity bar with SVG and text.



Choose a complex renderer and customize it.
Complex renderers can be customzied based on some data model properties. You can also add or remove decorations that can be customized.

7. Click **Finish**.

8. With the new rule selected under Style Rules:

   1. In the Styling Customizer select the **Graphic** tab

   2. Customize the Renderers as follows:

      a. Select the SVG element and browse to select an SVG file of your choice.

      b. Select the Text element and modify the default value for the Label to the value of the turnaround time instead of the value of the name, by entering `@turnaround`.

9. To save the new chart:

   1. Click **File > Save As**.

10. Enter the name `harbor2` for the project.

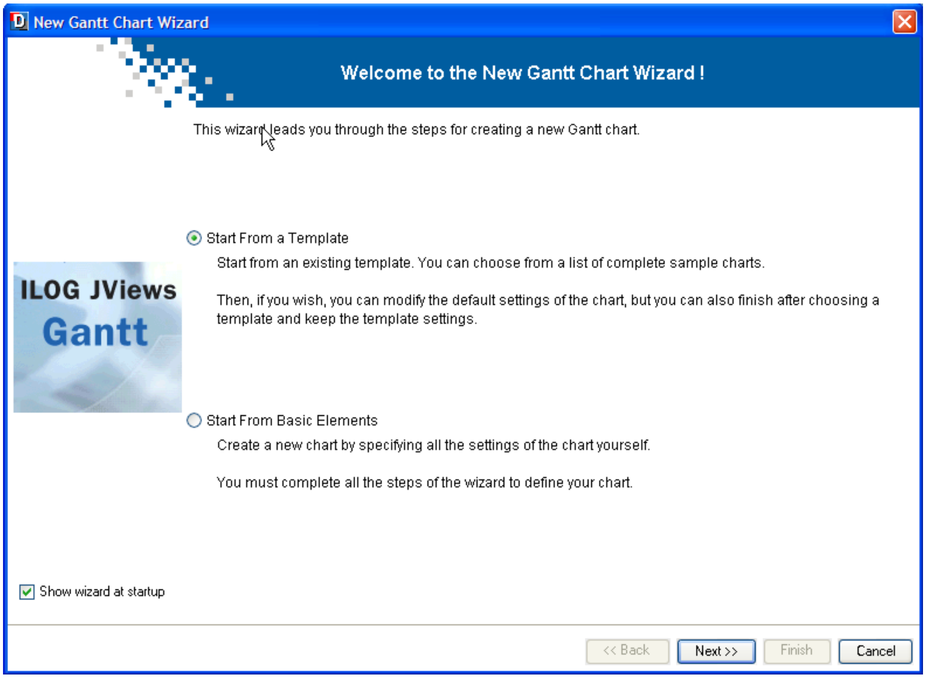
Creating Gantt charts from templates

A template is a predefined project with a style sheet designed for a specific type of application or style of presentation. It includes an appropriate data model and some sample activities, resources, constraints, and reservations with user-defined types. You can use a template as a starting point for a Gantt chart.

When you use the New Gantt Chart Wizard, you can create a Gantt chart by loading your data and applying basic settings, or you can load a template.

1. To start a new chart based on a supplied template, choose **File > New from Wizard**. The New Gantt Chart Wizard starts.

   The following figure shows how to choose the template option.



This wizard leads you through the steps for creating a new Gantt chart.

● Start From a Template

Start from an existing template. You can choose from a list of complete sample charts.

Then, if you wish, you can modify the default settings of the chart, but you can also finish after choosing a template and keep the template settings.

○ Start From Basic Elements

Create a new chart by specifying all the settings of the chart yourself.

You must complete all the steps of the wizard to define your chart.

☑ Show wizard at startup

2. Select the option to start from a predefined template.

   The following templates are available:

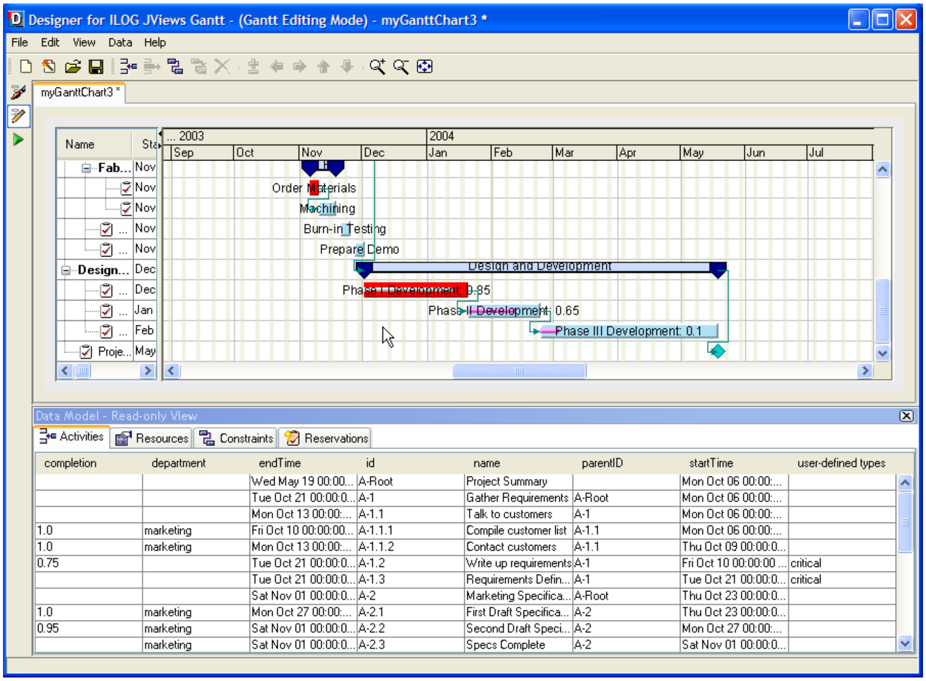
   - Basic Gantt Chart
   - Basic Schedule Chart
   - Gantt Status Chart
   - Schedule Chart for Plant PowerOps
   - Schedule Chart for Transportation Planning

These templates reflect typical uses of JViews Gantt. Each may have different styling options and different default styles for data.

3. Click **Next** and choose the **Gantt Status Chart**.

   The Gantt Status Chart template is an enhanced activity-oriented chart which includes the `completion` attribute in the data and the user-defined type `critical`. If you click **Finish**, you will have an activity-oriented chart that displays critical activities in red and shows the percentage completion as a narrow bar along the center of the activity bar.

   The following figure shows an enhanced activity-oriented chart with template data.



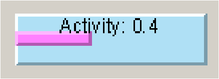
completion	department	endTime	id	name	parentID	startTime	user-defined types
		Wed May 19 00:00...	A-Root	Project Summary		Mon Oct 06 00:00:...	
		Tue Oct 21 00:00:0...	A-1	Gather Requirements	A-Root	Mon Oct 06 00:00:...	
		Mon Oct 13 00:00:...	A-1.1	Talk to customers	A-1	Mon Oct 06 00:00:...	
1.0	marketing	Fri Oct 10 00:00:00...	A-1.1.1	Compile customer list	A-1.1	Mon Oct 06 00:00:...	
1.0	marketing	Mon Oct 13 00:00:...	A-1.1.2	Contact customers	A-1.1	Thu Oct 09 00:00:0...	
0.75		Tue Oct 21 00:00:0...	A-1.2	Write up requirements	A-1	Fri Oct 10 00:00:00 ...	critical
		Tue Oct 21 00:00:0...	A-1.3	Requirements Defin...	A-1	Tue Oct 21 00:00:0...	critical
		Sat Nov 01 00:00:0...	A-2	Marketing Specifica...	A-Root	Thu Oct 23 00:00:0...	
1.0	marketing	Mon Oct 27 00:00:...	A-2.1	First Draft Specifica...	A-2	Thu Oct 23 00:00:0...	
0.95	marketing	Sat Nov 01 00:00:0...	A-2.2	Second Draft Speci...	A-2	Mon Oct 27 00:00:...	
	marketing	Sat Nov 01 00:00:0...	A-2.3	Specs Complete	A-2	Sat Nov 01 00:00:0...	

Note that the chart is shown in Chart Editing Mode with the data model read-only view.

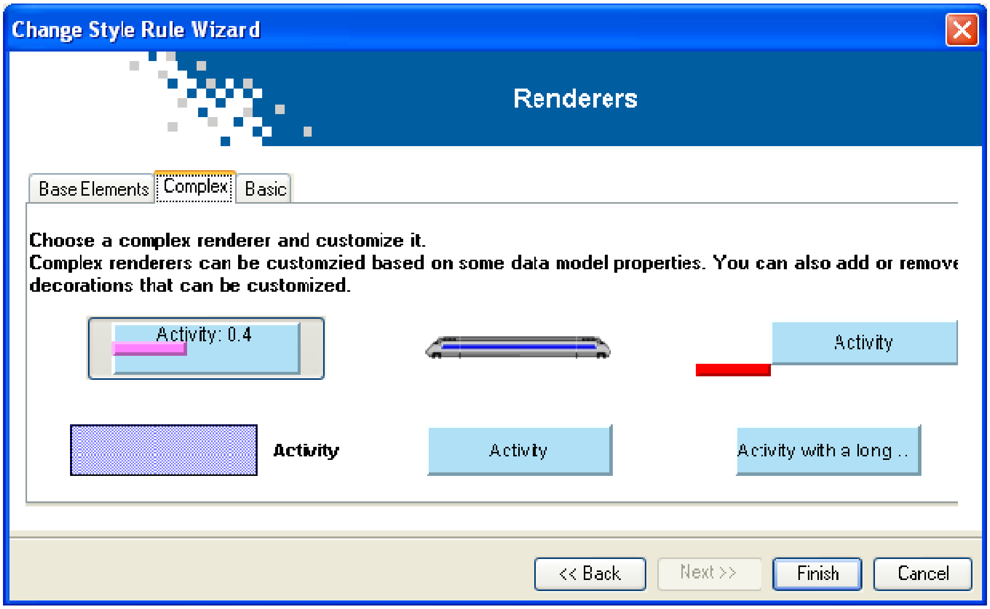
Using predefined complex renderers

JViews Gantt provides prebuilt complex renderers made up of composite graphic objects. These complex renderers can be customized. They include a complex renderer supplied for use with data that includes the completion property as in the `monitoringGantt` example used in *Getting Started*.

The following figure shows an activity bar with completion.



The following figure shows the complex renderers.



Choose a complex renderer and customize it.
Complex renderers can be customzied based on some data model properties. You can also add or remove decorations that can be customized.

This use case is based on the `monitoringGantt` example that you extended in *Getting Started*.

1. Select the style rule `activity.critical:leaf.engineering.completion`.

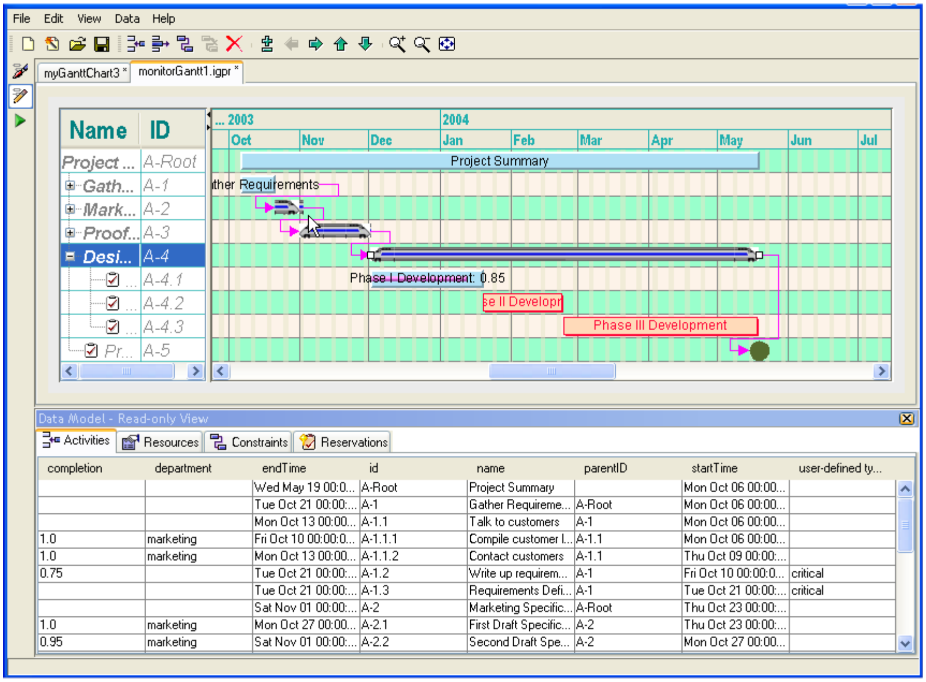
   You created this style rule in *Create a style rule condition for an activity with a user-defined type*.

2. Right-click and choose **Change Style Rule**.
3. Confirm that you want to modify the style rule by clicking **Yes**.
4. Clear **Generated Name** and type the original name of this style rule.
5. Click **Next** to leave the condition unchanged and access the **Renderer Mapping** panel.

6. Select **New renderer** and click **Next**.
7. Select **Complex** renderers.
8. Select the activity bar with completion shown previously.

   You can see the result in the Gantt chart if you switch into Gantt Chart Editing Mode.

   The following figure shows an activity bar rendering used in conjunction with completion property.



completion	department	endTime	id	name	parentID	startTime	user-defined ty...
		Wed May 19 00:0...	A-Root	Project Summary		Mon Oct 06 00:00...	
		Tue Oct 21 00:00:...	A-1	Gather Requireme...	A-Root	Mon Oct 06 00:00...	
		Mon Oct 13 00:00...	A-1.1	Talk to customers	A-1	Mon Oct 06 00:00...	
1.0	marketing	Fri Oct 10 00:00:0...	A-1.1.1	Compile customer I...	A-1.1	Mon Oct 06 00:00...	
1.0	marketing	Mon Oct 13 00:00...	A-1.1.2	Contact customers	A-1.1	Thu Oct 09 00:00:...	
0.75		Tue Oct 21 00:00:...	A-1.2	Write up requirem...	A-1	Fri Oct 10 00:00:0...	critical
		Tue Oct 21 00:00:...	A-1.3	Requirements Defi...	A-1	Tue Oct 21 00:00:...	critical
		Sat Nov 01 00:00:...	A-2	Marketing Specific...	A-Root	Thu Oct 23 00:00:...	
1.0	marketing	Mon Oct 27 00:00...	A-2.1	First Draft Specific...	A-2	Thu Oct 23 00:00:...	
0.95	marketing	Sat Nov 01 00:00:...	A-2.2	Second Draft Spe...	A-2	Mon Oct 27 00:00...	

Creating a complex renderer

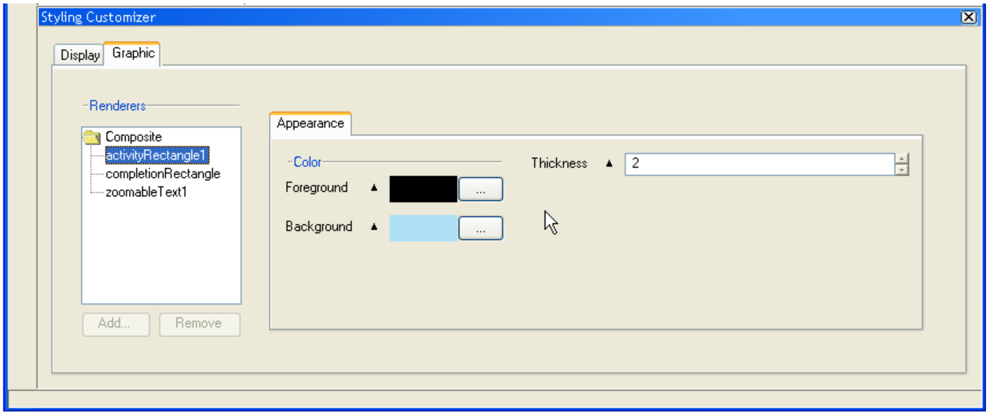
Explains how to create a complex renderer using composite graphics.

In this section

Overview

In this section you will learn how to create a complex renderer by using composite graphics. In the Designer, you use composite graphics to represent the activities in the chart. Composite graphics are like a lego set. Each composite graphic is made up of elements. You can put together existing graphic objects to construct new ones.

The following figure shows the composition of a complex renderer.



Thickness ▲ 2

The `Composite` node in the hierarchy represents the envelope that contains all the graphic objects that make up the complex renderer.

The next node in the tree, `activityRectangle1`, is the base element to which all the other graphic objects that compose the complex renderer are attached. You cannot remove the base element.

The remaining nodes are graphic objects that are attached to the base element.

You can customize the attachments by playing with their properties. You can customize the complex renderer by adding new attachments and removing existing ones.

You must choose a graphic object as the base element of your composite to which to attach other graphic objects. The following base elements are available in the renderers of the Create Style Rule Wizard or Change Style Rule Wizard:

- Simple rectangle
- Rectangle with customizable paint

  The paint facilities allow you to achieve more sophisticated rendering.
- Shadow rectangle
- Relief rectangle

The initial object is just a starting point; when you exit the wizard you can customize the object by setting properties, selecting a different image file, adding elements, and so on. You cannot delete the base element of a composite graphic object.

Then you can add elements to the composite graphic object by attaching other graphic objects to the base element. You can choose from:

- Icon: A raster image file (`.gif`, `.jpg`, or `.png`)
- Shape: A geometric shape
- SVG: A vector image file (`.svg`)
- Text: A text consisting of one or more lines

The example is based on `projectResources.igpr` located in:

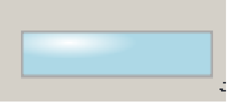
`<installdir>/jviews-gantt86/bin/designer/data/examples`

In this section you will learn about:

- *Selecting a base element*.
- *Styling the base element*.
- *Adding attachments*.

Selecting a base element

You must select a base element shape to which to attach the other graphic objects to compose the complex renderer.

1. Select `activity:leaf`.
2. Right-click and choose **Change Style Rule**.
3. Select **Yes**.
4. Clear **Generated Name** and type the original name of this style rule.
5. Click **Next** to leave the condition unchanged and access the **Renderer Mapping** panel.
6. Select **New renderer** and click **Next**.
7. Select **Base Elements**.
8. Select Rectangle with customizable paint  and click **Finish**.

Styling the base element

You can change the shape of the selected base element and apply more sophisticated styling to the fill color and pattern with the Paint feature.

1. Select the rule `activity:leaf` and click the **Graphic** tab in the Styling Customizer.

   You can see the container of the composite graphic with the base element of the graphic identified as `activityShape`.

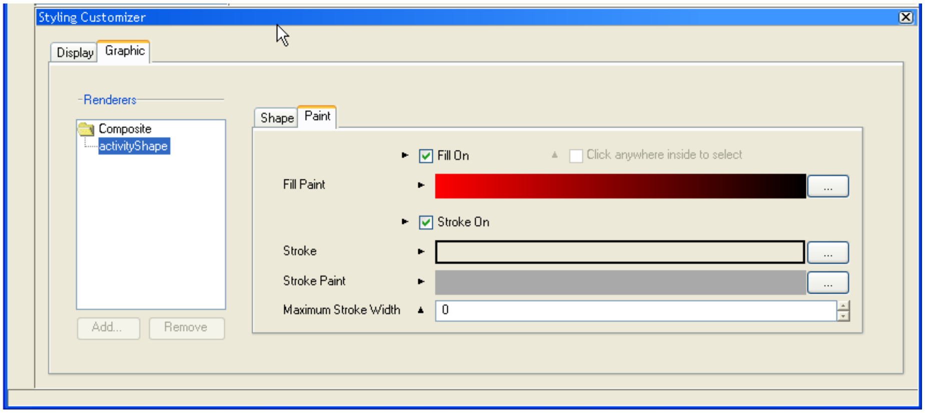
   The base element initially selected is a rectangle. You can change the shape by selecting a different geometric shape. However, the size of the activity shape is adjusted to the length of time that the activity it represents lasts. Therefore, the rectangular shape is particularly suited to this type of representation. You cannot change the shape of graphic objects that form attachments to the base element.

   Since the shape of the base element is adjusted to the duration of the activity that it represents, the base element of a milestone, which is a zero-duration activity, is not displayed, regardless of any width or height settings in the Styling Customizer. The base element of a complex renderer of milestones serves only as the base to accept attachments, such as an icon. See *Applying composite rendering to milestones*.

2. Click the **Paint** tab and click the ellipsis (...) in the Fill Paint field to access the Paint Editor.

3. Choose and **Apply** a different color and a linear gradient.

   The following figure shows the paint styling of the base element of a complex renderer.



☑ Fill On

□ Click anywhere inside to select

☑ Stroke On

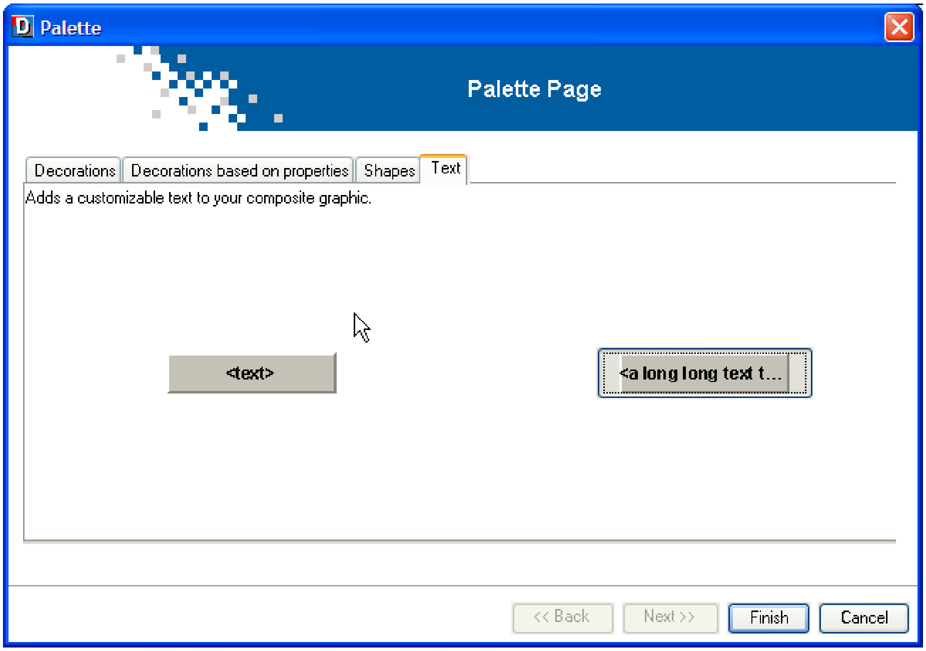
Maximum Stroke Width 0

Adding attachments

You can add attachments when the container of the composite graphic objects is selected.

1. Select **Composite** and click **Add**.
2. In the **Palette**, select the **Text** tab.
3. Select the long text that stays inside its container and click **Finish**.

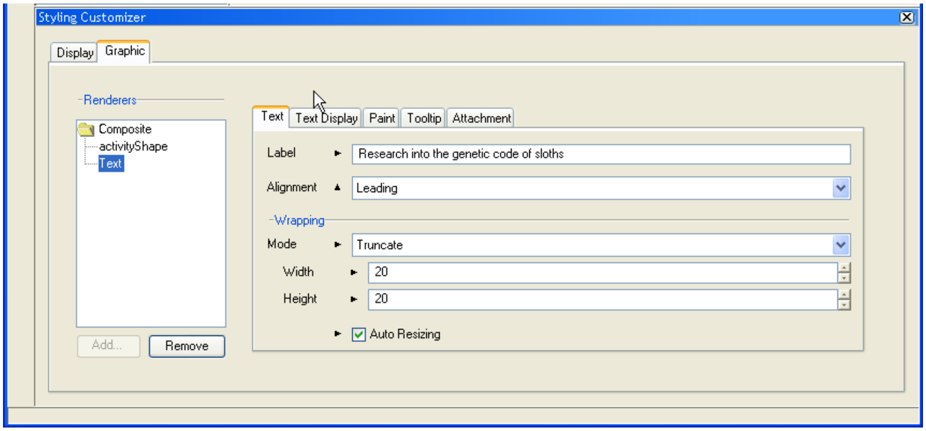
   The following figure shows the palette for text in composite graphics.



Adds a customizable text to your composite graphic.

4. Select **Text** in the tree of composite graphic object attachments to style the attachment.
5. Select the **Text** tab and type a long text into the **Label** field and leave the rest of the settings unchanged.

   The following figure shows how to style a text attachment.



- Composite
  - activityShape
  - Text

Label ▸ Research into the genetic code of sloths

Alignment ▴ Leading

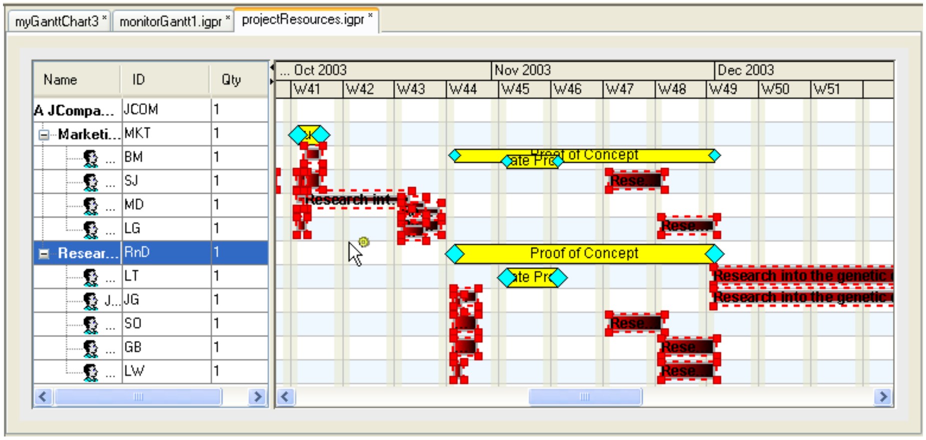
Mode ▸ Truncate

Width ▸ 20

Height ▸ 20

▸ ☑ Auto Resizing

The following figure shows the result.



Name	ID	Qty
A JCompa...	JCOM	1
Marketi...	MKT	1
...	BM	1
...	SJ	1
...	MD	1
...	LG	1
Resear...	RnD	1
...	LT	1
J...	JG	1
...	SO	1
...	GB	1
...	LW	1

Experiment with the other Text property settings to see the effect in the chart. Refer to Styling the Resource Data chart data in **Developing with the SDK** for information on what the different properties are for. You can see the property names by choosing **View > Styling Properties** in the Designer.

6. Select the container **Composite** and click **Add**.

7. In the **Palette** select the **Decorations** tab.

8. Select an icon and click **Finish**.

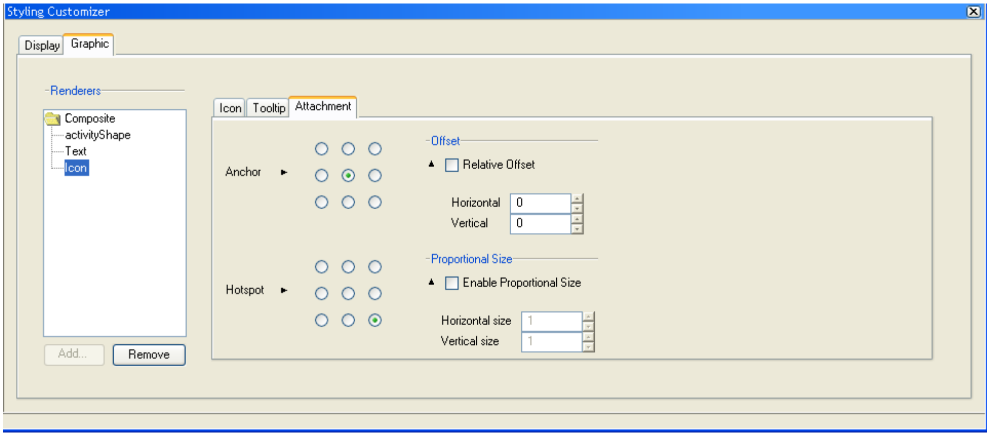
   If you want to use your own icon, you must first complete the modifications to the rule. Then, with the rule selected, you can browse in the Styling Customizer to select the URL of the raster image file of the icon you want to use.

9. Select **Icon** in the tree of composite graphics objects.

10. Select the **Attachment** tab and adjust the Hotspot attachment point to the bottom right.

You need to specify two attachment points, one on the base element, the anchor, and one on the attachment itself, the hotspot. The position of the attachment point is specified in relation to left, right, or center, top, bottom, or center by means of the matrix selection.

The following figure shows how to attach an icon.



Anchor ►
○ ○ ○
○ ● ○
○ ○ ○

Hotspot ►
○ ○ ○
○ ○ ○
○ ○ ●

▲ □ Relative Offset

Horizontal 0
Vertical 0

▲ □ Enable Proportional Size

Horizontal size 1
Vertical size 1

You can specify an icon to be used as a decoration by clicking the **Icon** tab and selecting the image location in the file browser.

You can remove a decoration by clicking **Remove**.

Applying composite rendering to milestones

In this example, you are going to attach an SVG graphic to the milestones in the chart. Open the project file that contains your customized `monitoringGantt` example.

1. Select the style rule `activity:milestone`.
2. Right-click and choose **Change Style Rule**.
3. Select **Yes**.
4. Clear **Generated Name** and type the original name of this style rule.
5. Click **Next** to leave the condition unchanged and access the **Renderer Mapping** panel.
6. Select **New renderer** and click **Next**.
7. Select **Base Elements**.
8. Select **Simple rectangle** and click **Finish**.

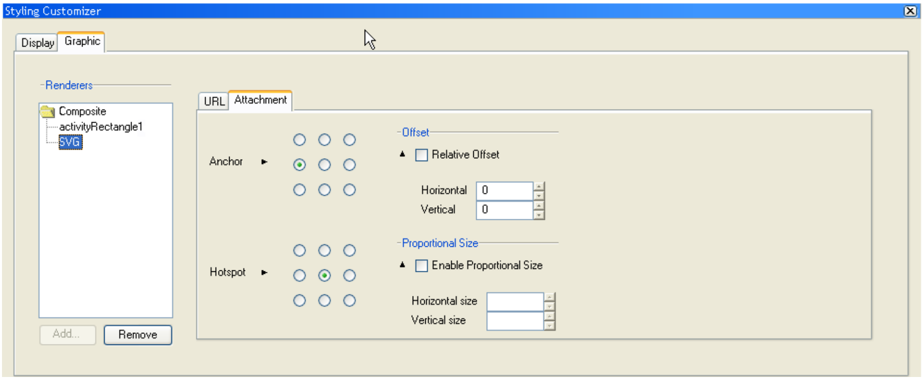
When the Graphic tab is selected in the Styling Customizer you can see that the base element `activityRectangle1` has been added to the container of the composite graphic objects. Since milestones have zero duration, you cannot see the shape used to form the base element. There is no point in styling it by color or shape, since it will not be displayed in the chart. However, it must exist so that you can attach a visible item to it.

1. Select **Composite** and click **Add**.
2. In the **Palette** select the **Decorations** tab.
3. Select the **graphic with SVG** and click **Finish**.

   If you want to use your own SVG file you must first complete the modifications to the rule. Then, with the rule selected, you can browse in the Styling Customizer to select the URL of the SVG file you want to use.

4. Select **SVG** in the tree of composite graphics objects.
5. Select the **Attachment** tab and make sure that the anchor point is set to center left.

   The following figure shows the SVG attachment for milestone.


Styling Customizer

Display | Graphic

Renderers
- Composite
  - activityRectangle1
  - SVG

Add... | Remove

URL | Attachment

Anchor ▸
○ ○ ○
● ○ ○
○ ○ ○

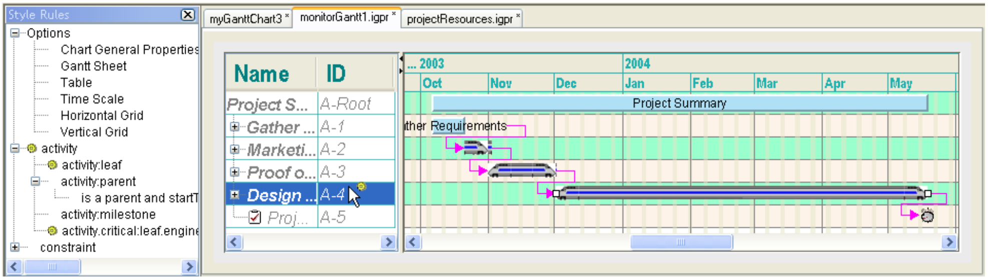
Hotspot ▸
○ ○ ○
○ ● ○
○ ○ ○

Offset
▴ □ Relative Offset
Horizontal 0
Vertical 0

Proportional Size
▴ □ Enable Proportional Size
Horizontal size
Vertical size

You must not center the anchor. Milestones have zero duration, so the Designer cannot calculate the center and therefore cannot display the graphic.

The following figure shows the result.


Style Rules
- Options
  - Chart General Properties
  - Gantt Sheet
  - Table
  - Time Scale
  - Horizontal Grid
  - Vertical Grid
- activity
  - activity:leaf
  - activity:parent
    - is a parent and startT
  - activity:milestone
  - activity.critical:leaf.engine
- constraint

myGanttChart3 * | monitorGantt1.igpr * | projectResources.igpr *

Name	ID
Project S...	A-Root
Gather ...	A-1
Marketi...	A-2
Proof o...	A-3
Design ...	A-4
Proj...	A-5

You can specify an SVG graphic to be used as a decoration by clicking the **URL** tab and entering the graphic location.

Writing an application

Shows you how to prepare an application to accept and deploy a Gantt chart component.

In this section

Overview

This section shows you how to integrate a Gantt chart component developed in the Designer into a Java® application. You will need to write some Java code to do this. The information in this section expands on the information in *Integrating a Gantt chart into an application*. It shows you how to prepare an application to accept and deploy a Gantt chart component.

Preparing to deploy a Gantt chart component

To build and deploy a JViews Gantt chart component, you must first consider the manner of deployment and the functional type of the application.

A JViews Gantt chart application can be deployed as:

- A Swing application
- Eclipse/RCP application
- A Thin Client
- An applet

This section discusses deployment as a Swing application.

An Eclipse/RCP application with JViews Gantt is implemented like a Swing application. Additional considerations are explained in Using JViews products in Eclipse RCP applications in **Developing with the SDK**.

See Deploying an application as a DHTML-only thin client in *Building Web Applications* and Deploying as an applet in *Developing with the SDK* for how to deploy an application as a Thin Client or an applet.

Creating a basic application from JViews Gantt beans

The basic steps for creating an application that will use a Gantt or Schedule chart are:

1. Create your GUI with various Swing components, such as a working area, a menu bar and a toolbar.

```
public class SimpleGantt extends JPanel
{
  public IlvHierarchyChart chart;
  public SimpleGantt(JFrame frame) {
    super();

    // Prepare the JPanel's appearance
    setLayout(new BorderLayout());
```

2. Create an instance of a Gantt or Schedule chart in the Java® API.

```
IlvGanttChart chart = new IlvGanttChart();
```

3. Connect the instance to a data model or load a project file that has been created and saved in the Designer.

```
// Bind the Gantt Chart to the Data Model
IlvGanttModel model = new IlvDefaultGanttModel();
chart.setGanttModel(model);
```

   or

```
// Apply the project file to the chart
chart.setProject(project);
```

   See *Integrating a Gantt chart into an application* for more about loading a project file. There are sample project files in:

   `<installdir>/jviews-gantt86/bin/designer/data/examples`

> **Note**: After creating a project using the Designer for JViews Gantt with a JDBC data source, the test/preview mode does not allow editing of the activities. The same project is not editable in a running application either. This happens when you use `IlvGanttJDBCDataSource`, which will result in a read-only model.

   Instead of loading a project file, you can load a custom data model and then load a style sheet (.css file) defined in the Designer. See *Integrating a style sheet into an application*. In this case, you use only the styling developed with the Designer and not the whole project.

Refer to the samples in **`<installdir>/jviews-gantt86/samples`** provided with IBM® ILOG® JViews Gantt for examples of how to create applications that use Gantt or Schedule charts.

The example **<installdir>/jviews-gantt86/codefragments/application/simpleGantt/src/SimpleGantt.java** shows a Swing application with a basic Gantt chart. This Swing application extends JPanel. It is located in:

```
<installdir>/jviews-gantt86/codefragments/application/simpleGantt
```

See Gantt charts in **Developing with the SDK** for how to create a Gantt chart and customize it with the SDK instead of the Designer.

XML structure

Describes the XML elements used to define scheduling data in the Schedule Data eXchange Language (SDXL) and the XML structure of a Designer project file.

In this section

Overview
Describes the contents of the following sections.

Content and structure of an XML data file
Describes the properties and uses of the Schedule Data eXchange Language (SDXL) elements.

XML structure of a Designer project file
Describes the properties of a Designer project file.

Overview

This section describes the XML elements used to describe scheduling data by reference to the file `monitorGantt.xml` used in *Getting Started*. You can see how the Designer loads data from this file in *Loading data from an Existing Gantt XML file*.

This section also describes the XML elements used to define a Designer project file.

Content and structure of an XML data file

Describes the properties and uses of the Schedule Data eXchange Language (SDXL) elements.

In this section

Overview

Data from an XML data source must comply with the DTD used in the Schedule Data eXchange Language (SDXL). See Document type definition for SDXL in **Developing with the SDK**.

The XML file used in the examples in *Getting Started* is located in:

```
<installdir>/jviews-gantt86/bin/designer/data/examples/monitorGantt.xml
```

You can learn more in general about SDXL in The schedule data exchange language in *Developing with the SDK*.

The schedule element

The schedule element is the root element that contains all the other elements in the XML document. It describes the content of a specific Gantt data model. It therefore contains:

- An activities element
- A resources element
- A constraints element
- A reservations element

Defining the root in XML

```
<schedule version="5.5">
  <title>Scheduling Data</title>
  <desc>ILOG JViews Scheduling Data Exchange Language</desc>
  ...
</schedule>
```

The schedule element has the following attribute:

version	%Text	Required	The version of the SDXL.

The title and desc elements are used to specify extra descriptive information about the schedule element.

The property element

The `property` element is used to describe user-defined properties added to objects in the Gantt data model. See *User-defined properties* for more information about how the `property` element is used.

User-defined types in XML

```
<activity end="21-10-2003 0:0:0" id="A-1.2"
  name="Write up requirements" start="10-10-2003 0:0:0">
  <property javaClass="java.lang.Float" name="completion">0.75</property>
  <property name="tags">critical</property>
</activity>
```

The `property` element has the following attributes:

`name`	`%Text`	Required	The name of the property.
`javaClass`	`%Text`	Implied	The property class name.

Resources

Resources belong to a Schedule chart. A `resources` element, the root of the resources, groups all `resource` elements in a `schedule` element. In other words, there is only one `resources` element in a `schedule` element.

A resource element corresponds to a resource object in a Gantt data model.

Defining resources in XML

```
<resources>
  <resource id="JCOM" name="A JCompany Employees" quantity="1.0">
    <resource id="MKT" name="Marketing" quantity="1.0">
      <resource id="BM" name="Bill McDonald" quantity="1.0"/>
      <resource id="SJ" name="Steve Knoll" quantity="1.0"/>
      <resource id="MD" name="Michael Smith" quantity="1.0"/>
      <resource id="LG" name="Luc Dupont" quantity="1.0"/>
    </resource>
    <resource id="RnD" name="Research and Development" quantity="1.0">
      <resource id="LT" name="Linus Dane" quantity="1.0"/>
      <resource id="JG" name="James Hook" quantity="1.0"/>
      <resource id="SO" name="Scott Washington" quantity="1.0"/>
      <resource id="GB" name="Grady Happy" quantity="1.0"/>
      <resource id="LW" name="Larry Money" quantity="1.0"/>
    </resource>
  </resource>
</resources>
```

A `resource` element is defined by the following attributes:

`id`	`ID`	Required	The identifier of a resource must be unique within the set of all activities and resources in the schedule.
`name`	`%Text`	Required	The name of the resource.
`quantity`	`%Text`	Implied	The number of resources.

A `resource` element can have any number of `property` elements.

In the example, the parent resource is defined as the department to which the individual leaf resources, the workers, belong.

Activities

Activities belong to a Gantt chart. An `activities` element, the root of the activities, groups all `activity` elements in a `schedule` element. In other words, there is only one `activities` element in a `schedule` element.

```
<activities dateFormat="d-M-yyyy H:m:s">
    <activity end="19-5-2004 0:0:0" id="A-Root" name="Project Summary"
                                                start="6-10-2003 0:0:0">
      <activity end="21-10-2003 0:0:0" id="A-1"
        name="Gather Requirements" start="6-10-2003 0:0:0">
        <activity end="13-10-2003 0:0:0" id="A-1.1"
          name="Talk to customers" start="6-10-2003 0:0:0">
          <activity end="10-10-2003 0:0:0" id="A-1.1.1"
            name="Compile customer list" start="6-10-2003 0:0:0">
            <property javaClass="java.lang.Float" name="completion">
                    1.0</property>
            <property name="department">marketing</property>
          </activity>
          <activity end="13-10-2003 0:0:0" id="A-1.1.2"
            name="Contact customers" start="9-10-2003 0:0:0">
            <property javaClass="java.lang.Float" name="completion">
                    1.0</property>
            <property name="department">marketing</property>
          </activity>
        </activity>
        <activity end="21-10-2003 0:0:0" id="A-1.2"
          name="Write up requirements" start="10-10-2003 0:0:0">
          <property javaClass="java.lang.Float" name="completion">
                    0.75</property>
          <property name="tags">critical</property>
        </activity>
        <activity end="21-10-2003 0:0:0" id="A-1.3"
          name="Requirements Defined" start="21-10-2003 0:0:0">
          <property name="tags">critical</property>
        </activity>
      </activity>
      <activity end="1-11-2003 0:0:0" id="A-2"
        name="Marketing Specification" start="23-10-2003 0:0:0">
        <activity end="27-10-2003 0:0:0" id="A-2.1"
          name="First Draft Specification" start="23-10-2003 0:0:0">
          <property javaClass="java.lang.Float" name="completion">
                    1.0</property>
          <property name="department">marketing</property>
        </activity>
        <activity end="1-11-2003 0:0:0" id="A-2.2"
          name="Second Draft Specification" start="27-10-2003 0:0:0">
          <property javaClass="java.lang.Float" name="completion">
                    0.95</property>
          <property name="department">marketing</property>
        </activity>
        <activity end="1-11-2003 0:0:0" id="A-2.3" name="Specs Complete"
                   start="1-11-2003 0:0:0">
```

```
  <property name="department">marketing</property>
 </activity>
</activity>
```

Defining activities in XML

```
<activity end="2-12-2003 0:0:0" id="A-3" name="Proof of Concept"
       start="1-11-2003 0:0:0">
 <activity end="14-11-2003 0:0:0" id="A-3.1" name="Rough Design"
       start="1-11-2003 0:0:0">
  <activity end="8-11-2003 0:0:0" id="A-3.1.1" name="CAD Layout"
       start="1-11-2003 0:0:0">
   <property javaClass="java.lang.Float" name="completion">
       0.3</property>
  </activity>
  <activity end="14-11-2003 0:0:0" id="A-3.1.2" name="Detailing"
       start="8-11-2003 0:0:0">
   <property javaClass="java.lang.Float" name="completion">
       0.7</property>
  </activity>
 </activity>
 <activity end="18-11-2003 0:0:0" id="A-3.2"
  name="Fabricate Prototype" start="6-11-2003 0:0:0">
  <activity end="10-11-2003 0:0:0" id="A-3.2.1"
   name="Order Materials" start="6-11-2003 0:0:0">
   <property name="tags">critical</property>
  </activity>
  <activity end="18-11-2003 0:0:0" id="A-3.2.2" name="Machining"
       start="10-11-2003 0:0:0"/>
 </activity>
 <activity end="25-11-2003 0:0:0" id="A-3.3"
  name="Burn-in Testing" start="20-11-2003 0:0:0"/>
 <activity end="2-12-2003 0:0:0" id="A-3.4" name="Prepare Demo"
       start="27-11-2003 0:0:0"/>
</activity>
<activity end="19-5-2004 0:0:0" id="A-4"
 name="Design and Development" start="2-12-2003 0:0:0">
 <activity end="20-1-2004 0:0:0" id="A-4.1"
  name="Phase I Development" start="2-12-2003 0:0:0">
  <property name="department">engineering</property>
  <property javaClass="java.lang.Float" name="completion">
       0.85</property>
  <property name="tags">critical</property>
 </activity>
 <activity end="24-2-2004 0:0:0" id="A-4.2"
  name="Phase II Development" start="20-1-2004 0:0:0">
  <property javaClass="java.lang.Float" name="completion">
       0.65</property>
  <property name="department">engineering</property>
 </activity>
 <activity end="19-5-2004 0:0:0" id="A-4.3"
  name="Phase III Development" start="24-2-2004 0:0:0">
  <property javaClass="java.lang.Float" name="completion">
       0.1</property>
  <property name="department">engineering</property>
```

```
        </activity>
      </activity>
      <activity end="19-5-2004 0:0:0" id="A-5" name="Project Complete"
                    start="19-5-2004 0:0:0"/>
    </activity>
  </activities>
```

The `dateFormat` attribute defines the date format used for activities.

An `activity` element corresponds to an activity object in a Gantt data model. It is used to define a simple activity and is defined by the following attributes:

`id`	`ID`	Required	The identifier of an activity must be unique within the set of all activities and resources in the schedule.
`name`	`%Text`	Required	The name of the activity.
`start`	`%Datetime`	Required	The start time of the activity.
`end`	`%Datetime`	Required	The end time of the activity.

An activity element can have any number of `property` elements.

An activity group is defined in the same way as a simple activity, but contains nested simple `activity` elements. It is the parent activity. In the example, the activity A-1, Gather Requirements, is a parent activity. It has a child activity A-1.1, Talk to customers, which has the leaf activities A-1.1.1, Compile customer list and A-1.1.2, Contact customers.

You can use the root, the activity group, and simple activities to define a hierarchy of tasks in the schedule.

User-defined properties

The `property` element is used to define your own properties that do not exist in the supplied Gantt data model. These properties can be referred to in style rules for styling your data. In the Style Rule Wizard, these properties are referred to as user-defined types. You write style rules in the Style Rule Wizard of the Designer to customize the styling of your data.

A `property` element named `completion` is used to represent the percentage completion of an activity. For example:

Use of the completion property element

```
        <activity end="24-2-2004 0:0:0" id="A-4.2"
          name="Phase II Development" start="20-1-2004 0:0:0">
            <property javaClass="java.lang.Float" name="completion">
                  0.65</property>
          <property name="department">engineering</property>
        </activity>
```

Later, when the `property` element `dueDate` is introduced, you can use the value of `completion` to determine whether an activity is on schedule, behind schedule, or even in advance.

The property element can be used to attribute an activity as a task to be performed by a resource, for example, the marketing department.

Distributing tasks to resources

```
        <activity end="1-11-2003 0:0:0" id="A-2.2"
          name="Second Draft Specification" start="27-10-2003 0:0:0">
          <property javaClass="java.lang.Float" name="completion">
                    0.95</property>
          <property name="department">marketing</property>
        </activity>
```

In *Distributing tasks to resources*, the `property` attribute named `"department"` is set to `marketing` for the activity `"Second Draft Specification"`.

The user-defined property with the reserved name `tags` is used to distinguish the `property` element used for a user-defined type that will be interpreted as a CSS class for styling purposes. *User-defined type in XML* shows how to define the user-defined type `critical`, which is used to distinguish the styling of activities that are in a critical state, so that they can be flagged visually in the schedule.

User-defined type in XML

```
        <activity end="18-11-2003 0:0:0" id="A-3.2"
          name="Fabricate Prototype" start="6-11-2003 0:0:0">
          <activity end="10-11-2003 0:0:0" id="A-3.2.1"
            name="Order Materials" start="6-11-2003 0:0:0">
            <property name="tags">critical</property>
          </activity>
```

Constraints

A constraints element, the root element, groups all constraint elements in a schedule element. In other words, there is only one constraints element in a schedule element:

Defining constraints in XML

```
<constraints>
  <constraint from="A-4.1" to="A-4.2" type="End-Start"/>
  <constraint from="A-4.2" to="A-4.3" type="End-Start"/>
  <constraint from="A-3.1.1" to="A-3.1.2" type="End-Start"/>
  <constraint from="A-1.1" to="A-1.2" type="End-Start"/>
  <constraint from="A-3.2.1" to="A-3.2.2" type="End-Start"/>
  <constraint from="A-2.1" to="A-2.2" type="End-Start"/>
  <constraint from="A-4" to="A-5" type="End-Start"/>
  <constraint from="A-2" to="A-3" type="End-Start"/>
  <constraint from="A-1" to="A-2" type="End-Start"/>
  <constraint from="A-1.1.1" to="A-1.1.2" type="End-Start"/>
  <constraint from="A-3" to="A-4" type="End-Start"/>
  <constraint from="A-2.2" to="A-2.3" type="End-Start"/>
  <constraint from="A-1.2" to="A-1.3" type="End-Start"/>
</constraints>
```

A constraint element can have any number of property elements.

Constraints are used to place dependencies on activities, such as the start of one activity is dependent on the completion of another activity. For this purpose, an activity is described as a From activity or a To activity.

A From activity is the source activity of a constraint. The start or end of such an activity controls the start or end of another activity.

A To activity is the target activity of a constraint. The start or end of such an activity controls the start or end of another activity as the result of the constraint.

In the example, all the constraints are defined as end-to-start constraints, which means that the end of the To activity depends on the start of the From activity.

A constraint element corresponds to a constraint object in a Gantt data model. It has the following attributes:

from	%ActivityID	Required	The identifier of the From activity, which must be a valid activity in the same SDXL file.
to	%ActivityID	Required	The identifier of the To activity, which must be a valid activity in the same SDXL file.
type	%ConstraintType	Required	The type of the constraint. It can be: Start-Start Start-End End-Start End-End

Reservations

A `reservations` element, the root element, groups all `reservation` elements in a `schedule` element. In other words, there is only one `reservations` element in a `schedule` element:

Defining reservations in XML

```
<reservations>
  <reservation activity="A-4.1" resource="LT"/>
  <reservation activity="A-3.2" resource="LT"/>
  <reservation activity="A-1.1.1" resource="SJ"/>
  <reservation activity="A-1.1.2" resource="BM"/>
  <reservation activity="A-1.1.1" resource="LG"/>
  <reservation activity="A-1.2" resource="MD"/>
  <reservation activity="A-3.4" resource="LG"/>
  <reservation activity="A-4.2" resource="LT"/>
  <reservation activity="A-2.1" resource="LG"/>
  <reservation activity="A-3.2" resource="BM"/>
  <reservation activity="A-4.2" resource="JG"/>
  <reservation activity="A-3.3" resource="SO"/>
  <reservation activity="A-4.3" resource="SO"/>
  <reservation activity="A-4.3" resource="LT"/>
  <reservation activity="A-1.1.1" resource="MD"/>
  <reservation activity="A-4.2" resource="SO"/>
  <reservation activity="A-4.3" resource="JG"/>
  <reservation activity="A-3.1.2" resource="SO"/>
  <reservation activity="A-2.2" resource="MD"/>
  <reservation activity="A-4.3" resource="LW"/>
  <reservation activity="A-3.1.2" resource="JG"/>
  <reservation activity="A-3.1.1" resource="LW"/>
  <reservation activity="A-1.1.2" resource="SJ"/>
  <reservation activity="A-3" resource="RnD"/>
  <reservation activity="A-2.1" resource="MD"/>
  <reservation activity="A-3.1.2" resource="GB"/>
  <reservation activity="A-3.3" resource="SJ"/>
  <reservation activity="A-2.2" resource="LG"/>
  <reservation activity="A-3.4" resource="GB"/>
  <reservation activity="A-3.1.1" resource="JG"/>
  <reservation activity="A-3" resource="BM"/>
  <reservation activity="A-3.4" resource="LW"/>
  <reservation activity="A-4.1" resource="JG"/>
  <reservation activity="A-1.1" resource="MKT"/>
</reservations>
```

A reservation ties a resource to a specific activity.

A `reservation` element corresponds to a reservation object in a Gantt data model. It has the following attributes:

activity	%ActivityID	Required	The identifier of the activity, which must be a valid activity in the same SDXL file.
resource	%ResourceID	Required	The identifier of the resource, which must be a valid resource in the same SDXL file.

A reservation element can have any number of property elements.

XML structure of a Designer project file

Describes the properties of a Designer project file.

In this section

Overview

The Designer lets you save your work in three files:

- Project file
- Style sheet
- Data file

The project file defines the Gantt project configuration by specifying the type of the Gantt chart, the reference to the style sheet, and the type of data source. The project file has an `.igpr` extension.

The following sections describe the XML elements used in the project file.

The ganttConfiguration element

The `ganttConfiguration` element is the root element that contains all the other elements in the XML document.

Usage of the ganttConfiguration element

```
<ganttConfiguration type="...">
  <style url="..." />
  <dataSource class="..." />
</ganttConfiguration>
```

The following table shows the `ganttConfiguration` element.

Element	Attribute and Type	Required	Description
`ganttConfiguration`			Describes the content of a specific Gantt project. It therefore contains a `style` element and a `dataSource` element.
	`type %Text`	Yes	The type of Gantt chart. The only values accepted are `activity` for a Gantt chart or `resource` for a Schedule chart.

The style element

The following table shows how the `style` element provides a reference to the style sheet associated with the project.

Element	Attribute and Type	Required	Description
`style`			Provides a reference to the style sheet associated with the project.
	`url %Text`	Yes	The URL of the style sheet.

The dataSource element

The following table shows how the dataSource element defines the type of data referenced by the project. This is done through the class attribute, which specifies the kind of data source to load.

Element	Attribute and Type	Required	Description
dataSource			Defines the type of data referenced by the project.
	class %Text	Yes	The class that defines the data source to load. The only values accepted are: ilog.views.gantt.project. IlvGanttXMLDataSource that reads from an XML file. IlvGanttJDBCDataSourceilog.views.gantt.project. IlvGanttJDBCDataSource that reads data from a database through the JDBC API.
	url %Text	Yes for IlvGanttXMLDataSource	Used for data sources of type IlvGanttXMLDataSource only to give the URL of the XML file.

When the data source is an IlvGanttJDBCDataSource, the data source contains different elements for the connection and for the queries. See *The connection element* and *The query element*.

Example of a basic project file

There are examples of project files in the directory:

`<installdir>/jviews-gantt86/bin/designer/data/examples`

The following code example shows the typical content of a project file.

Project file projectTasks.igpr

```
<?xml version="1.0" encoding="UTF-8"?>
<ganttConfiguration type="activity">
  <style url="projectTasks.css"/>
  <dataSource
    class="ilog.views.gantt.project.IlvGanttXMLDataSource"
    url="simpleProject.xml"/>
</ganttConfiguration>
```

The result of setting this project file on a Gantt chart creates a data model by loading the XML from the file `simpleProject.xml` and applies the style sheet `projectTasks.css` to it.

Elements for connecting to and querying a database

The elements described in this section define in the project file the information needed for connecting to and querying a database. The data obtained by the query is mapped to the Gantt data model.

The connection element

The following table shows how the `connection` element specifies a connection to a database. It can be used by `query` elements.

Element	Attribute and Type	Required	Description
`connection`			Specifies a connection to a database.
	`id %Text`	Yes	Provides a reference to the connection subsequently in the XML file.
	`url %Text`	Yes	Specifies the URL of the database.
	`user %Text`	No	Specifies the user for connecting to the database.
	`passwd %Text`	No	Specifies the password used for connecting to the database.
	`driver %Text`	Yes	Specifies the name of the JDBC driver class used for connecting to the database.
	`driverURL %Text`	No	Specifies the URL that points to a JAR file where the driver archive can be found. This attribute is **required** if the driver is not loaded directly by the application code.

The query element

The various `query` elements (on `activities`, `resources`, `reservations`, and `constraints`) define the queries needed to obtain the Gantt data model. See*Accessing the data* and also Connecting to data through JDBC in **Developing with the SDK**.

The following table shows the attributes of the query element.

Element	Attribute and Type	Required	Description
query			Defines a query needed to obtain the Gantt data model.
	connection %Text	Yes	Holds a reference to the ID of the connection as defined by the connection element illustrated in *The connection element*.
	value %Text	Yes	Defines an SQL query for obtaining results from the database.
	model %Text	Yes	Can be activities, resources, constraints, or reservations, depending on which table model the query is meant to fill.

The map element

The query elements accept map elements as child elements. The map elements provide the means of mapping the result of the SQL query to the Gantt data model properties. The map element itself is optional. When it is present, the query element recognizes the attributes of the map element and thus the result of the query can be mapped to the Gantt data model.

The following table shows the attributes of the map element.

Element	Attribute and Type	Required	Description
map			Maps the result of the SQL query to the Gantt data model.
	columns %Text	Yes	Provides the comma-separated list of columns in the SQL query from which the mapping is made.
	property %Text	Yes	Specifies the name of the property in the Gantt data model for the given activity, reservation, resource, or constraint to which the columns will be mapped.
	requiredType %Text	No	Specifies the Java type required by the property in the Gantt data model.

The value element

Optionally, the map element can accept child elements of type value for the constraintType property of the constraint object. This element specifies the type of constraint, which may be:

- START-START
- START-END
- END-START
- END-END

The following table shows the attributes of the value element.

Element	Attribute and Type	Required	Description
`value`			Specifies the type of constraint.
	`model %Text`	Yes	Specifies a value in the model for the type of constraint. For example: `ilog.views.gantt.IlvConstraintType.END_END`
	`table %Text`	Yes	A value in the result of the SQL query that should map to the value given by the `model` attribute.

Next steps after the Designer

This topic shows you how to adapt your component and data configuration by using the Java™ API.

It is based around the example of extending a project, which can be found at:

<installdir>/jviews-gantt86/codefragments/application/extendProject

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 `IlvGanttChart` and `IlvGanttConfiguration` classes.

Typical Java code for including the project file in an application

```
IlvGanttChart ganttChart = new IlvGanttChart()
ganttChart.setProject(url);
```

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

```
// Create a project configuration
IlvGanttProjectConfiguration configuration = new
IlvGanttProjectConfiguration();

// Read the configuration file.
configuration.read(url);

// The project configuration now contains the styling, the
//reference to the data source and the Gantt model from this
//data source.

// In order to change the data source of this configuration,
// we need to:
// 1. Change the data source of this configuration.
// 2. Create the data model from this data source, and put it
//     into the configuration.
// 3. Apply this new configuration to the Gantt

// 1. Change the data source by referencing a new XML file
IlvGanttXMLDataSource dataSource = new
                                           IlvGanttXMLDataSource();
dataSource.setDataURL(getResourceURL(complexXML));
```

```
configuration.setDataSource(dataSource);

// 2. Update the Gantt model of the configuration associated
// with this data source.

dataSource.read(configuration.getGanttModel());

// 3. Apply the new configuration to the chart.
configuration.apply(chart);
```

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 `IlvGanttModel` interface.

Developing with the JViews Gantt SDK presents these different concepts in detail and guides you through different options and steps.

Index