IBM ILOG Gantt for .NET V4.0

Tutorials

June 2009

© Copyright International Business Machines Corporation 1987, 2009.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

Preface Tutorials. 3

Creating a Windows Forms Gantt Application 1

Creating the Gantt Chart 1

Associating the Gantt Data Model With the Gantt Chart 10

Filling the Gantt Chart With XML Data 11

Creating a Web Forms Gantt Application 15

Creating the Gantt Chart 15

Associating the Gantt Data Model With the Gantt Chart 20

Filling the Gantt Chart With XML Data 20

Adding Controls to Interact with the Gantt Chart 22

Adding Controls to Modify the Gantt Data Model 23

Saving Modifications Across Requests. 23

Synchronizing a Schedule Chart with a Load Chart (Windows Forms Tutorial) 27

Creating a Schedule Chart 27

Associating a Gantt Data Model With the Schedule Chart. 36

Filling the Schedule Chart With XML Data. 38

Adding a Load Chart 42

Synchronizing the Charts **46**

Synchronizing the Visible Time Range 46

Synchronizing the Grids 47

Synchronizing the Charts Splitter 48

Synchronizing the Resource Selection 49

Making Reservations Visible at Launch Time **51**

Creating a Custom Gantt Representation (Windows Forms Tutorial) **55**

Creating the Activity Table **55**

Associating a Gantt Model With the Activity Table **64**

Filling the Activity Table With XML Data **66**

Adding the Scroll Activity Sheets. **70**

Making Reservation Visible at Launch Time **77**

Index **1**

Tutorials

The IBM® ILOG® Gantt for .NET tutorials provide step-by-step introductions to the fundamental areas of programming for the IBM ILOG Gantt for .NET framework. Each tutorial provides a detailed walkthrough of the steps required to write, build and run small sample applications.

In This Section

Creating a Windows Forms Gantt Application

Walks you through the process of creating a simple Gantt application using Windows® Forms.

Creating a Web Forms Gantt Application

Walks you through the process of creating a simple Gantt application using Web Forms.

Synchronizing a Schedule Chart with a Load Chart (Windows Forms Tutorial)

Walks you through the process of synchronizing a Schedule chart with a Load chart.

Creating a Custom Gantt Representation (Windows Forms Tutorial)

Walks you through the process of creating a custom Gantt chart representation made of an activity table and two scroll activity sheets.

Related Sections

Getting Started

Provides a hands-on introduction and overview to the IBM ILOG Gantt for .NET framework.

Creating a Windows Forms Gantt Application

In this tutorial you will learn how to create a simple Windows® Forms Gantt application.

See Also *Creating the Gantt Chart | Associating the Gantt Data Model With the Gantt Chart | Filling the Gantt Chart With XML Data*

Creating the Gantt Chart

The Gantt chart is one of the controls that IBM® ILOG® Gantt for .NET provides to display scheduling information. The Gantt chart is divided in three parts:

- A table view, an instance of GanttTable, which displays activity information from the data model.
- A Gantt Sheet (an instance of the class GanttSheet) which displays how activities are positioned on the time scale.
- A zoomable time scale, which is an instance of the class TimeScale.

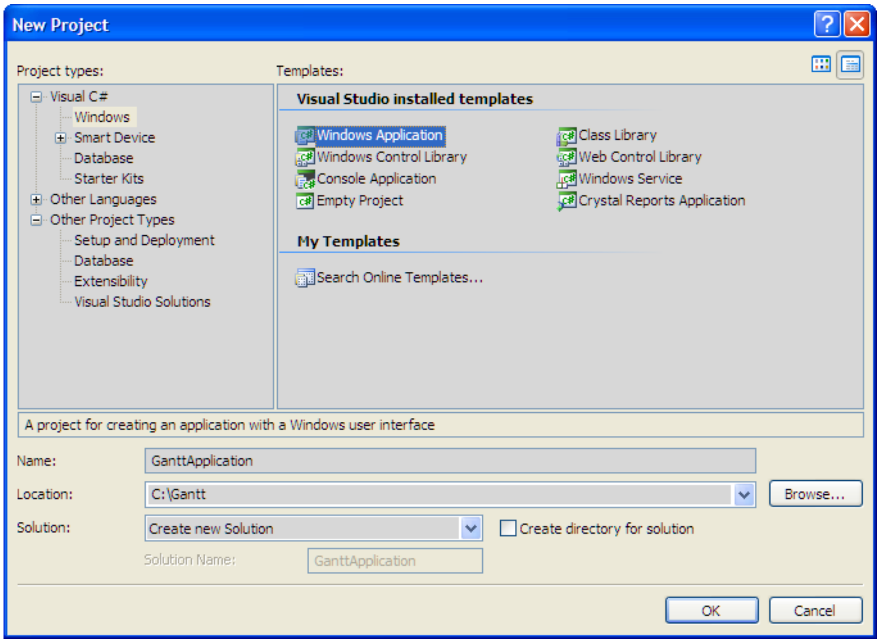
A standard, adjustable splitter separates the left part from the right part.

To create the Gantt application, proceed as follows:

1. Create a project of type Windows® Forms.

   On the **File** menu, click **New**, and then click **Project**.

   The New Project window appears.



A project for creating an application with a Windows user interface

Name: GanttApplication

Location: C:\Gantt

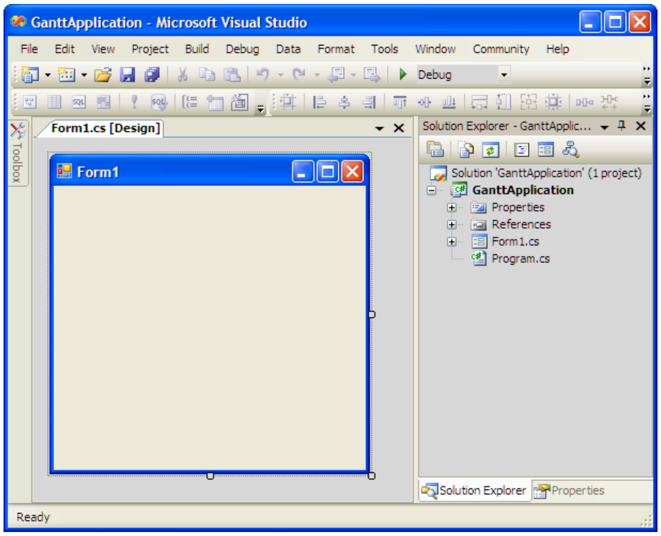
Solution: Create new Solution

Create directory for solution

Solution Name: GanttApplication

2. In the left window, click **Visual C#** > **Windows**.
3. In the right window, click **Windows Application**.
4. Name the project GanttApplication.
5. Place the project in **C:\Gantt**.

   Visual Studio®.NET creates a form named Form1 and stores it by default in the file **Form1.cs**.

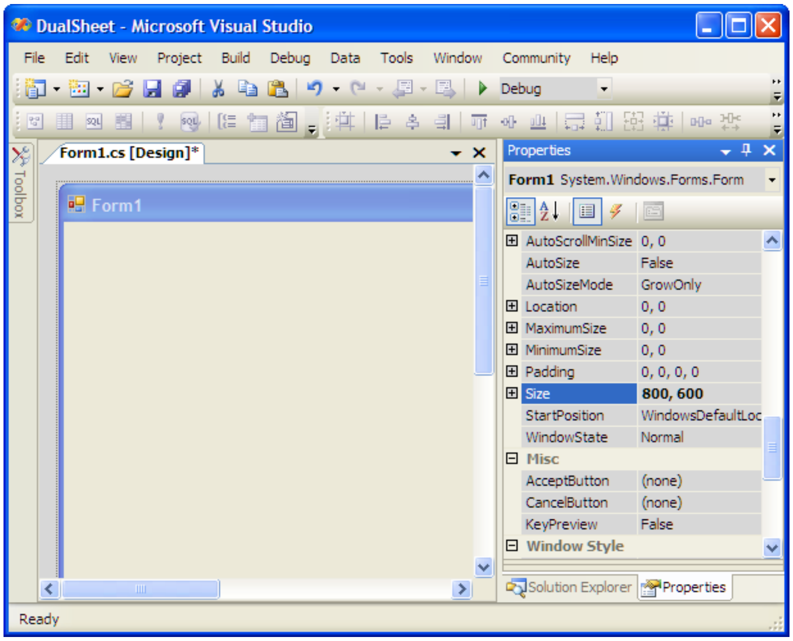

GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]
Form1
Solution Explorer - GanttApplic...
Solution 'GanttApplication' (1 project)
GanttApplication
Properties
References
Form1.cs
Program.cs
Solution Explorer
Properties
Ready

6. Resize the window.

From the menu bar, select View > Properties Window and from the Properties explorer choose **Layout > Size**. Expand the property and enter the following values:

Width: 800

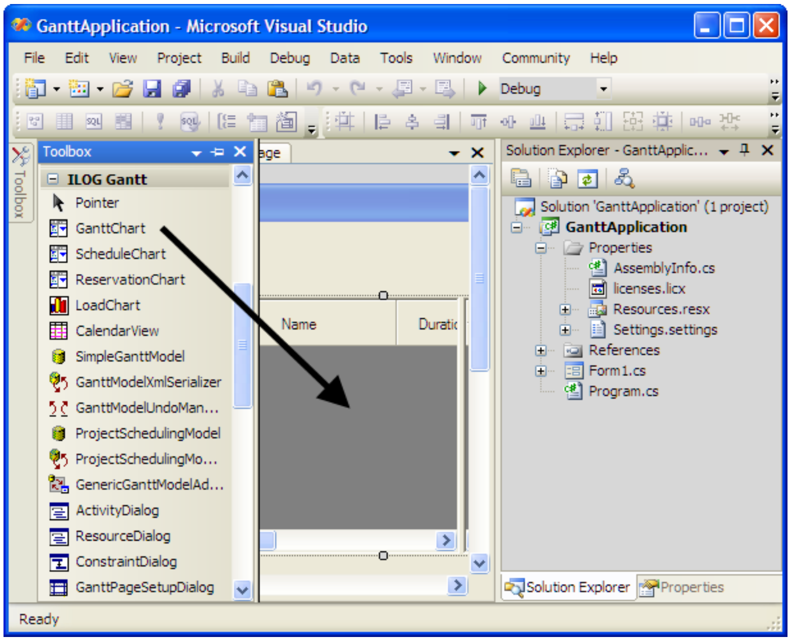
Height: 600


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Properties
Solution Explorer
Properties
Ready

Form1 System.Windows.Forms.Form

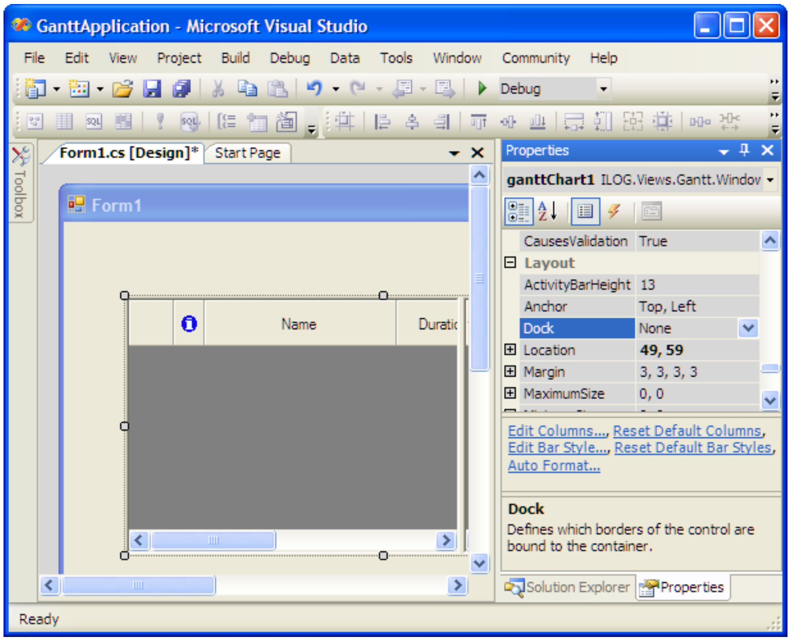
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
Location	0, 0
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	**800, 600**
StartPosition	WindowsDefaultLoc
WindowState	Normal
Misc	
AcceptButton	(none)
CancelButton	(none)
KeyPreview	False
Window Style	

7. From the IBM ILOG Gantt for .NET toolbox select the **GanttChart** object and drag it to Form1.



8. Change the Dock property to make the Gantt chart occupy the entire form.

Click the Gantt chart in Form1and from the Properties explorer choose **Layout** > **Dock**.

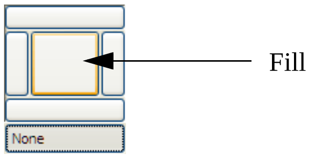

GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Start Page
Form1
Name
Duratic
Properties
ganttChart1 ILOG.Views.Gantt.Windov
Solution Explorer
Properties
Ready

CausesValidation	True
⊟ Layout	
ActivityBarHeight	13
Anchor	Top, Left
Dock	None
⊞ Location	49, 59
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0

Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...

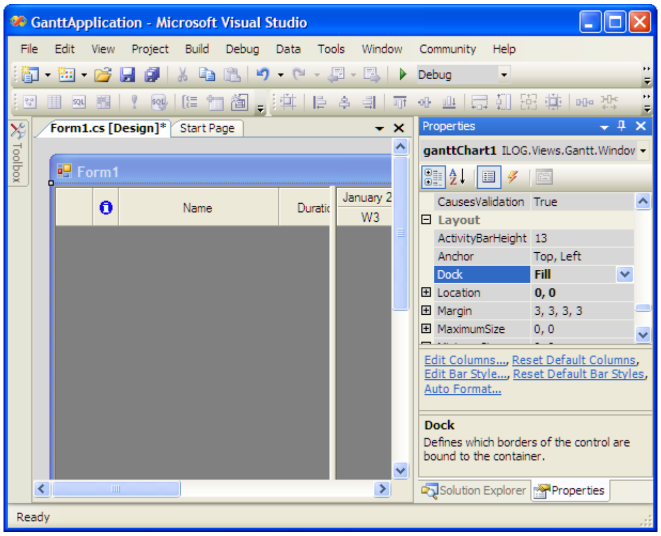
Dock
Defines which borders of the control are bound to the container.

Clicking the arrow next to the Dock property shows the following control:



9. Select the area corresponding to the Fill option.

Form1 now looks like this:


GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Start Page
Form1
Name
Duratic
January 2
W3
Properties
Solution Explorer
Properties
Ready

ganttChart1 ILOG.Views.Gantt.Window

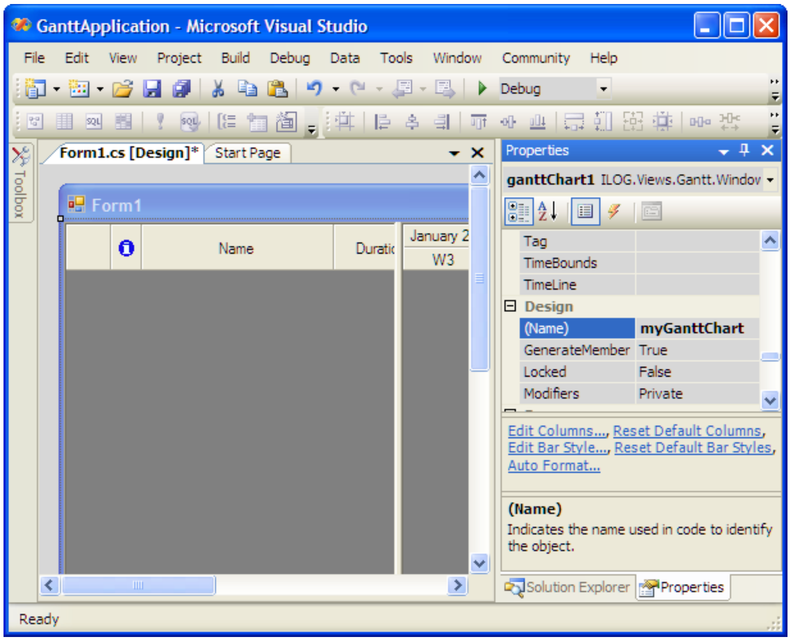
CausesValidation	True
Layout	
ActivityBarHeight	13
Anchor	Top, Left
Dock	Fill
Location	0, 0
Margin	3, 3, 3, 3
MaximumSize	0, 0

Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...

Dock
Defines which borders of the control are bound to the container.

10. Rename the Gantt chart.

Click the Gantt chart in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **myGanttChart**.


GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]* Start Page
Form1
Name
Duratic
January 2
W3
Properties
ganttChart1 ILOG.Views.Gantt.Window
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

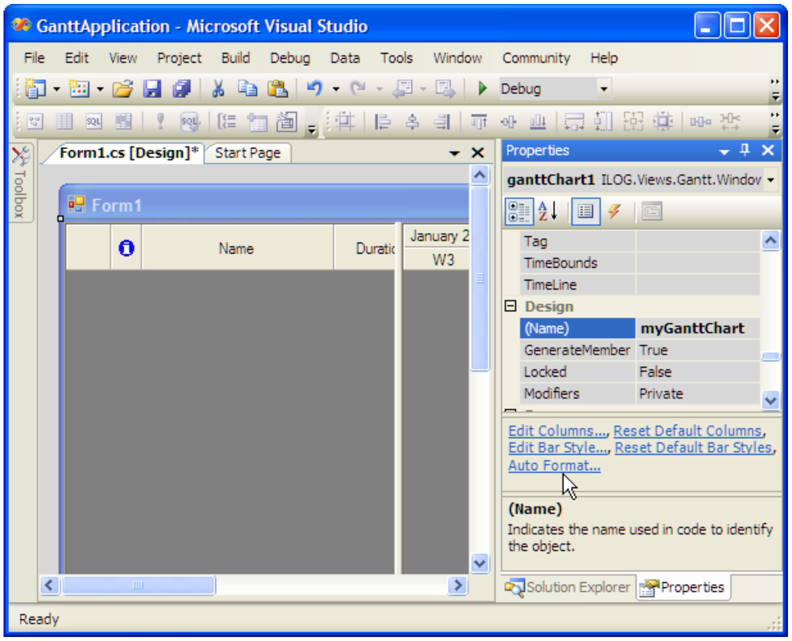
Tag	
TimeBounds	
TimeLine	
Design	
(Name)	**myGanttChart**
GenerateMember	True
Locked	False
Modifiers	Private

(Name)
Indicates the name used in code to identify the object.

The appearance of the Gantt chart can be completely customized to fit your needs.

11. Define a style for the Gantt chart you have just created.

You can either right-click the Gantt chart in Form1 and select **AutoFormat** in the contextual menu, or click **AutoFormat** at the bottom of the Properties explorer.


GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]* Start Page
Form1
Name
Duratic
January 2
W3
Properties
Solution Explorer Properties
Ready

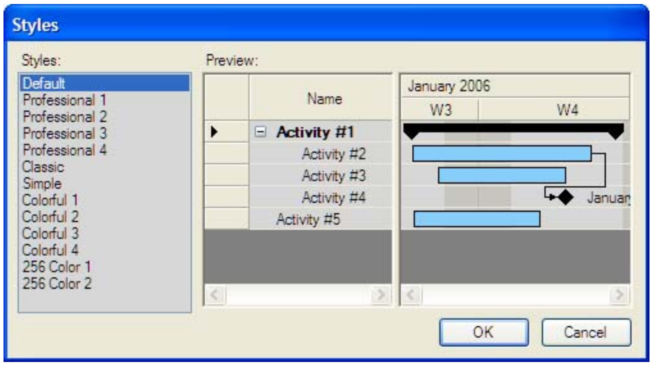
ganttChart1 ILOG.Views.Gantt.Windov

Tag	
TimeBounds	
TimeLine	
Design	
(Name)	**myGanttChart**
GenerateMember	True
Locked	False
Modifiers	Private

Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...

(Name)
Indicates the name used in code to identify the object.

The Styles window appears:



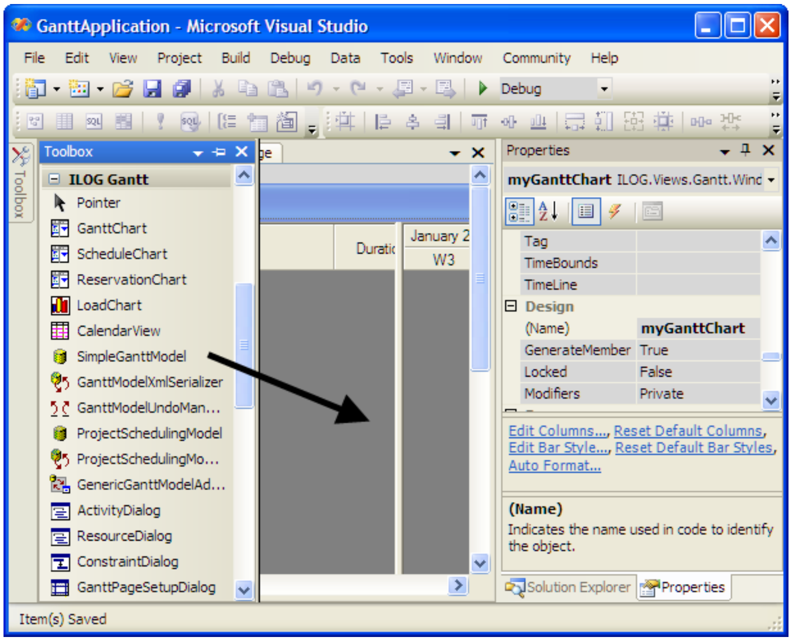
Select a style and click OK. The style that you selected is automatically applied to the Gantt chart.

12. Save the file **Form1.cs**.

Associating the Gantt Data Model With the Gantt Chart

The Gantt data model contains the scheduling information you want to display in the chart. In the following steps, you will learn how to bind a simple Gantt data model to the Gantt chart.

1. From the IBM® ILOG® Gantt for .NET toolbox select the **SimpleGanttModel** object and drag it to Form1.



(Name)	myGanttChart
GenerateMember	True
Locked	False
Modifiers	Private

(Name)
Indicates the name used in code to identify the object.

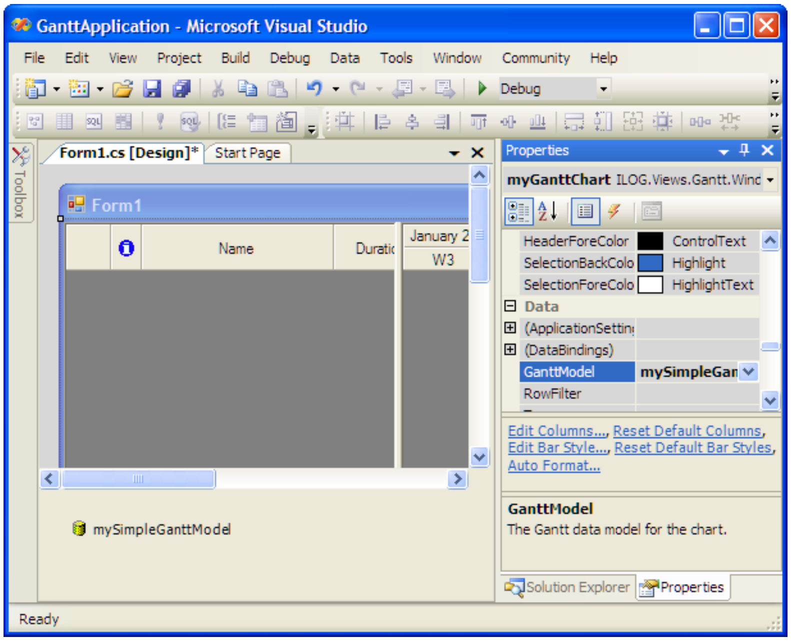
2. Rename the simple Gantt model.

   Click the **SimpleGanttModel1** object in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **mySimpleGanttModel**.

3. Associate **mySimpleGanttModel** with **myGanttChart**.

   Click the Gantt chart in Form1 and from the Properties explorer select **Data > GanttModel**.

   Click the arrow of the drop-down list next to the GanttModel property and choose **mySimpleGanttModel**.


GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]* Start Page
Form1
Name
Duratic
January 2
W3
mySimpleGanttModel
Properties
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

myGanttChart ILOG.Views.Gantt.Winc

HeaderForeColor	■ ControlText
SelectionBackColo	🟦 Highlight
SelectionForeColo	□ HighlightText
⊟ Data	
⊞ (ApplicationSettin	
⊞ (DataBindings)	
GanttModel	mySimpleGan
RowFilter	

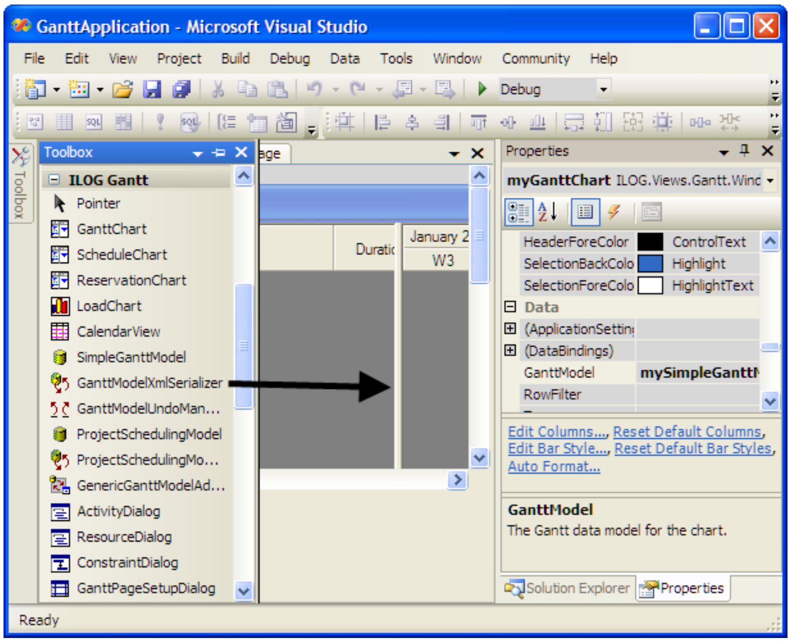
GanttModel
The Gantt data model for the chart.

Filling the Gantt Chart With XML Data

IBM® ILOG® Gantt for .NET allows you to serialize and deserialize scheduling data to and from Scheduling Data Exchange Language (SDXL) files. This is accomplished through the **GanttModelXmlSerializer** class.

To deserialize scheduling data from SDXL files and use this information to fill the Gantt chart, perform the following steps.

1. From the IBM ILOG Gantt for .NET toolbox select the **GanttModelXmlSerializer** object and drag it to Form1.



myGanttChart ILOG.Views.Gantt.Winc

HeaderForeColor	■ ControlText
SelectionBackColo	■ Highlight
SelectionForeColo	□ HighlightText
Data	
(ApplicationSettin	
(DataBindings)	
GanttModel	**mySimpleGanttM**
RowFilter	

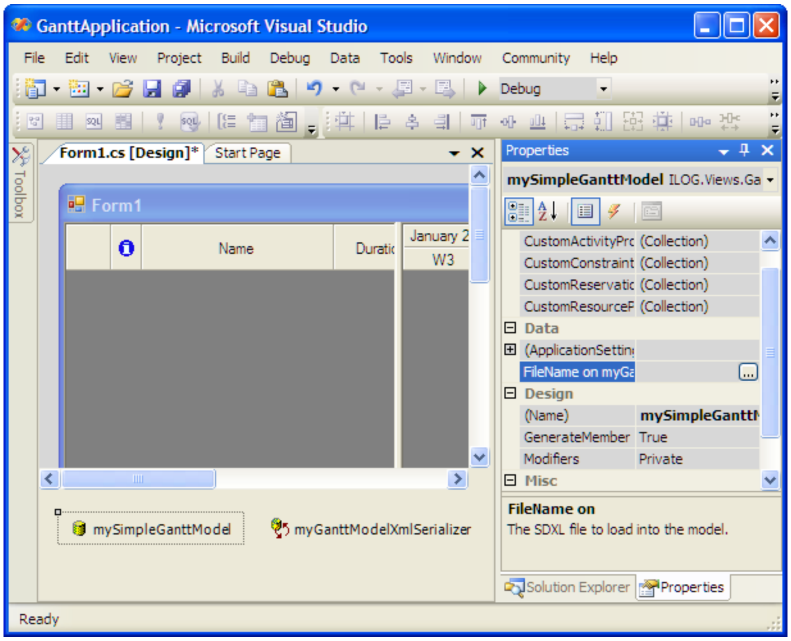
GanttModel
The Gantt data model for the chart.

2. Rename the Gantt model XML serializer.

Click the object **ganttModelXmlSerializer1** in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **myGanttModelXmlSerializer**.

3. Click **mySimpleGanttModel**.

You will notice that a new property named **FileName on myGanttModelXmlSerializer** has been created.

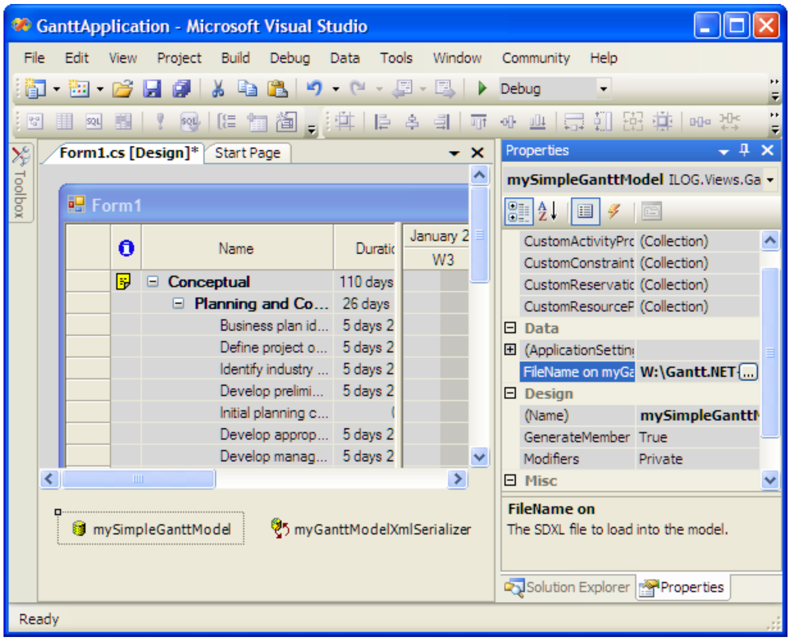

GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Start Page
Form1
Name
Duratic
January 2
W3
mySimpleGanttModel
myGanttModelXmlSerializer
Properties
mySimpleGanttModel ILOG.Views.Ga
Solution Explorer
Properties
Ready

CustomActivityPrc	(Collection)
CustomConstraint	(Collection)
CustomReservatic	(Collection)
CustomResourceF	(Collection)
Data	
(ApplicationSettin	
FileName on myGa	
Design	
(Name)	mySimpleGanttI
GenerateMember	True
Modifiers	Private
Misc	

FileName on
The SDXL file to load into the model.

4. Select the property **FileName on myGanttModelXmlSerializer** and click the three-dots button next to it. Choose an SDXL file to fill the Gantt chart.

Note: *Examples of SDXL files can be found in <install-dir>\Data.*


GanttApplication - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]* Start Page
Form1
January 2
W3
Properties
Solution Explorer Properties
Ready

Name	Duratic
Conceptual	110 days
Planning and Co...	26 days
Business plan id...	5 days 2
Define project o...	5 days 2
Identify industry ...	5 days 2
Develop prelimi...	5 days 2
Initial planning c...	(
Develop approp...	5 days 2
Develop manag...	5 days 2

mySimpleGanttModel ILOG.Views.Ga

CustomActivityPrc	(Collection)
CustomConstraint	(Collection)
CustomReservatic	(Collection)
CustomResourceF	(Collection)
Data	
(ApplicationSettin	
FileName on myGa	W:\Gantt.NET
Design	
(Name)	mySimpleGantt
GenerateMember	True
Modifiers	Private
Misc	

FileName on
The SDXL file to load into the model.

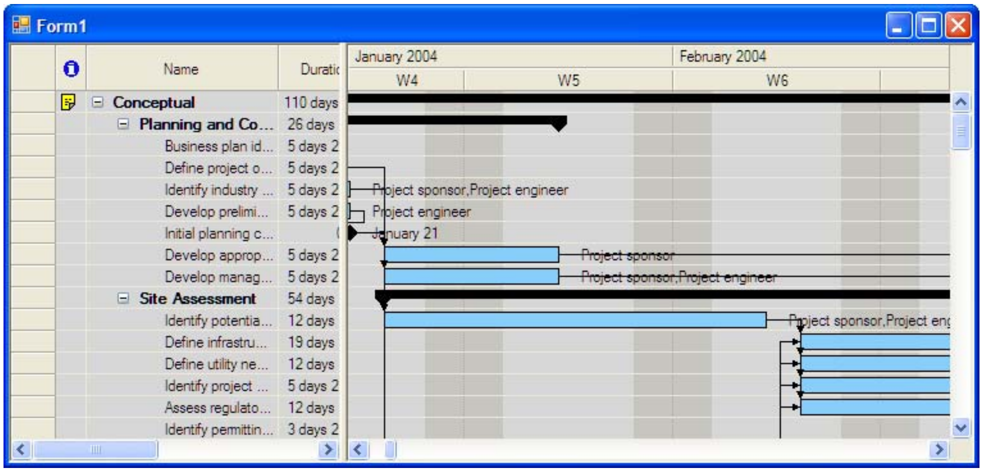
mySimpleGanttModel myGanttModelXmlSerializer

Data contained in the SDXL file are displayed in the Gantt chart.

5. Save the file **Form1.cs**.

6. Run the application using the command **Debug** > **Start**.

The Gantt application named Form1 should appear as follows:


Form1
January 2004
February 2004
W4
W5
W6
Project sponsor,Project engineer
Project engineer
January 21
Project sponsor
Project sponsor,Project engineer
Project sponsor,Project eng

Name	Duratic
Conceptual	110 days
Planning and Co...	26 days
Business plan id...	5 days 2
Define project o...	5 days 2
Identify industry ...	5 days 2
Develop prelimi...	5 days 2
Initial planning c...	(
Develop approp...	5 days 2
Develop manag...	5 days 2
Site Assessment	54 days
Identify potentia...	12 days
Define infrastru...	19 days
Define utility ne...	12 days
Identify project ...	5 days 2
Assess regulato...	12 days
Identify permittin...	3 days 2

Creating a Web Forms Gantt Application

In this tutorial you will learn how to create a simple Web Forms Gantt application.

See Also *Creating the Gantt Chart | Associating the Gantt Data Model With the Gantt Chart | Filling the Gantt Chart With XML Data | Adding Controls to Interact with the Gantt Chart | Adding Controls to Modify the Gantt Data Model | Saving Modifications Across Requests*

Creating the Gantt Chart

The Gantt chart is one of the controls that IBM® ILOG® Gantt for .NET provides to display scheduling data. The Gantt chart is divided in three parts:

- A table view, an instance of GanttTable, that displays activity information from the data model.
- A Gantt Sheet (an instance of GanttSheet) that displays how activities are positioned on the time scale.
- A zoomable time scale (an instance of TimeScale).

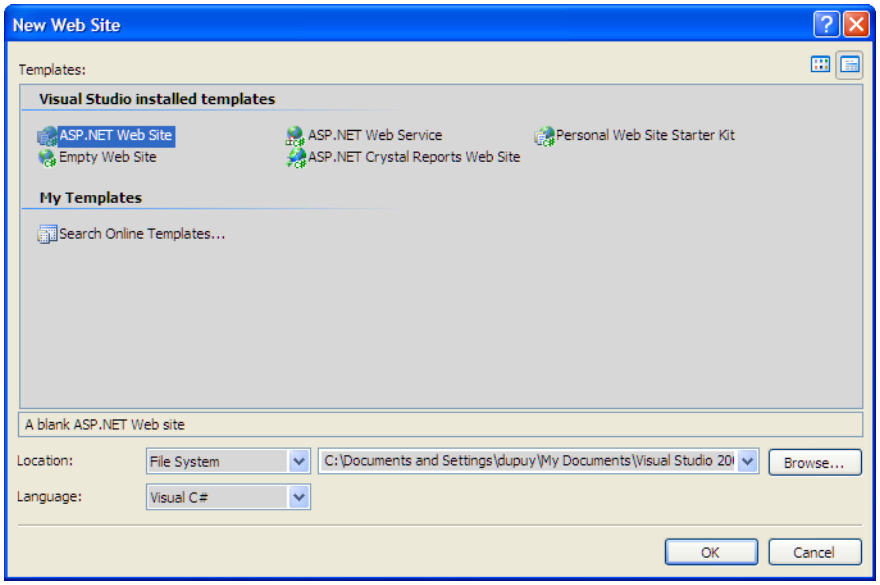
A standard, adjustable splitter separates the left part from the right part.

To create the Gantt application, proceed as follows:

1. Create a project of type ASP.NET Web Application.

On the **File** menu, click **New**, and then click **WebSite**.

The New Project window appears.


New Web Site
Templates:
Visual Studio installed templates
ASP.NET Web Site
ASP.NET Web Service
Personal Web Site Starter Kit
Empty Web Site
ASP.NET Crystal Reports Web Site
My Templates
Search Online Templates...
Browse...
OK
Cancel

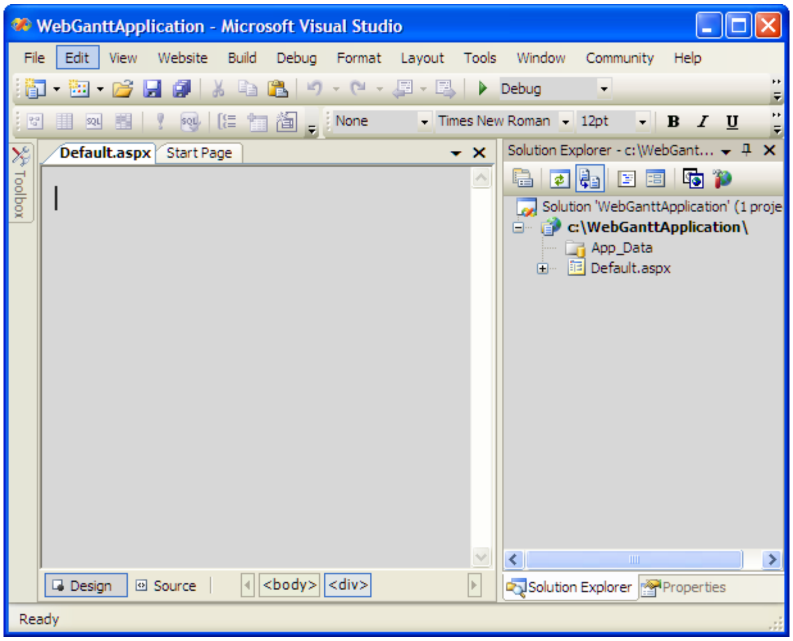
A blank ASP.NET Web site

Location: File System C:\Documents and Settings\dupuy\My Documents\Visual Studio 20

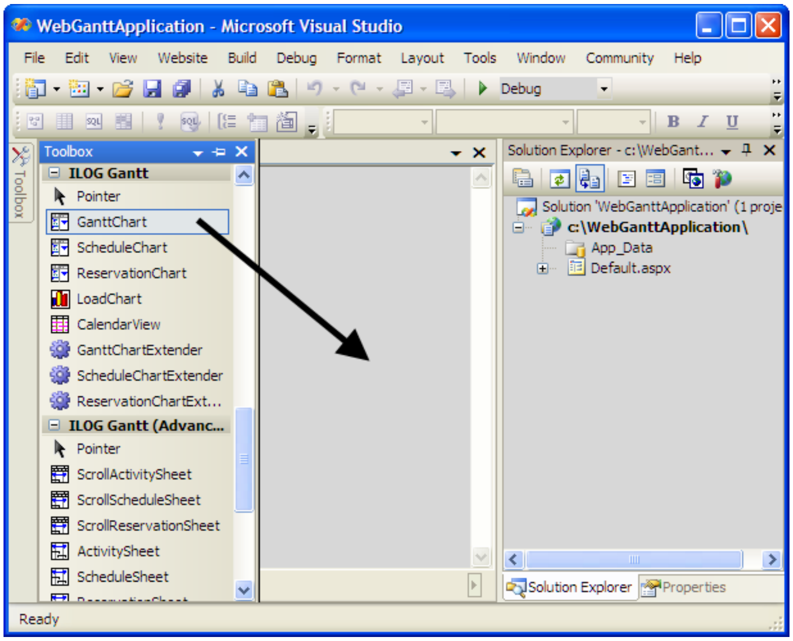
Language: Visual C#

2. Select **ASP.NET Web Site**.
3. Click in the location text box.
4. Name the project **C:\WebGanttApplication**.
5. Click the **OK** button.

   Visual Studio®.NET creates a empty form and stores it by default in the file **Default.aspx**.

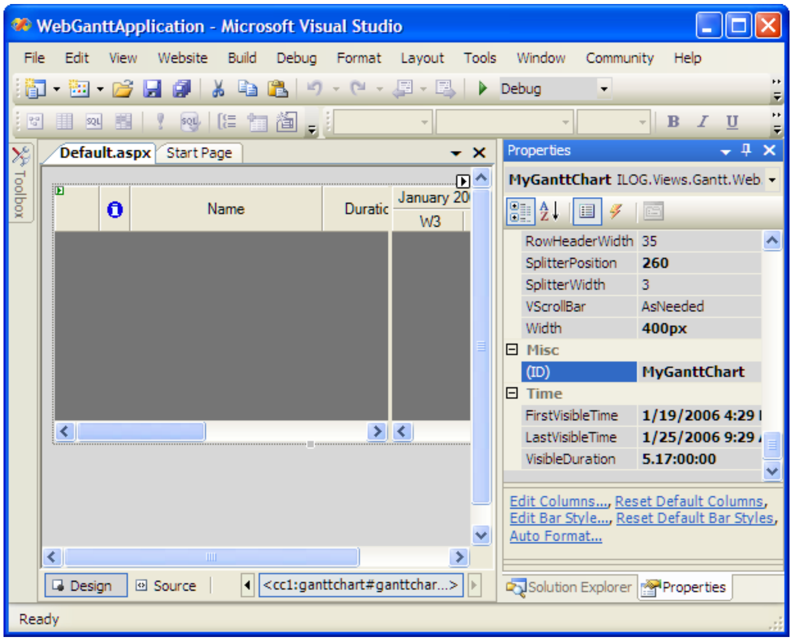

WebGanttApplication - Microsoft Visual Studio
File Edit View Website Build Debug Format Layout Tools Window Community Help
Debug
None
Times New Roman
12pt
Toolbox
Default.aspx
Start Page
Solution Explorer - c:\WebGant...
Solution 'WebGanttApplication' (1 proje
c:\WebGanttApplication\
App_Data
Default.aspx
Design
Source
<body>
<div>
Solution Explorer
Properties
Ready

6. From the IBM ILOG Gantt for .NET toolbox tab select the **GanttChart** object and drag it to the web form.



7. Rename the Gantt chart.

Select the Gantt chart in the web form. From the Properties explorer choose **Misc** > **ID** and enter the new name: **MyGanttChart**.


WebGanttApplication - Microsoft Visual Studio
File Edit View Website Build Debug Format Layout Tools Window Community Help
Debug
Default.aspx Start Page
Name Duratic January 20 W3
Design Source <cc1:ganttchart#ganttchar...>
Solution Explorer Properties
Ready

Properties

MyGanttChart ILOG.Views.Gantt.Web

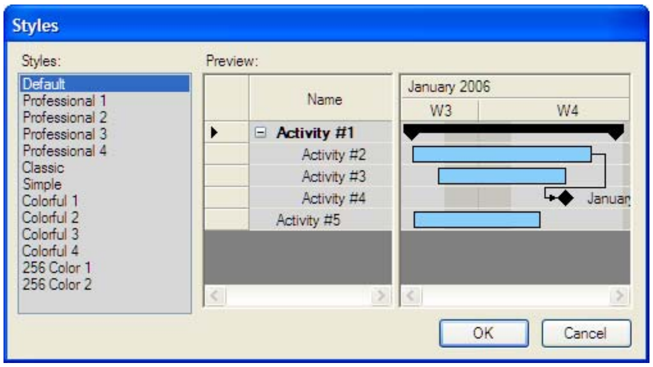
RowHeaderWidth	35
SplitterPosition	260
SplitterWidth	3
VScrollBar	AsNeeded
Width	400px
Misc	
(ID)	MyGanttChart
Time	
FirstVisibleTime	1/19/2006 4:29 I
LastVisibleTime	1/25/2006 9:29 /
VisibleDuration	5.17:00:00

Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...

8. Resize the Gantt chart to make it larger.

9. Define a style for the Gantt chart you have just created.

You can either right-click the Gantt chart in **Default** and select **AutoFormat** in the contextual menu, or click **AutoFormat** at the bottom of the Properties explorer. The Styles window appears:



Select a style and click **OK**. The style that you selected is automatically applied to the Gantt chart.

10. Save the file **Default.aspx.**

Associating the Gantt Data Model With the Gantt Chart

The Gantt data model contains the scheduling data you want to display in the chart. In the following steps, you will learn how to bind a simple Gantt data model to the Gantt chart.

1. Open the **Default.aspx.cs** file and adds the following variable declarations:

```
private ILOG.Views.Gantt.Data.SimpleGanttModel ganttModel;
```

2. Override the **OnInit** method to create the Gantt model:

```
protected override void OnInit(EventArgs e)
{
    ganttModel = new ILOG.Views.Gantt.Data.SimpleGanttModel;
    base.OnInit(e);
}
```

3. Override the **OnLoad** method to set the Gantt model of the Gantt chart:

```
protected override void OnLoad(EventArgs e)
{
    MyGanttChart.GanttModel = ganttModel;
    base.OnLoad(e);
}
```

Filling the Gantt Chart With XML Data

IBM® ILOG® Gantt for .NET allows you to serialize and deserialize scheduling data to and from Scheduling Data Exchange Language (SDXL) files. This is accomplished through the **GanttModelXmlSerializer** class.

To deserialize scheduling data from SDXL files and use this information to fill the Gantt chart, perform the following steps.

1. Open the **Default.aspx.cs** file and adds the following variable declaration:

```
private ILOG.Views.Gantt.Data.GanttModelXmlSerializer ganttSerializer;
```

2. Modify the **OnInit** method to create the serializer:

```
protected override void OnInit(EventArgs e)
{
    ganttModel = new ILOG.Views.Gantt.Data.SimpleGanttModel;
    ganttSerializer = new ILOG.Views.Gantt.Data.GanttModelXmlSerializer();
    base.OnInit(e);
}
```

3. Modify the **OnLoad** method to load a SDXL file into the Gantt model:

```
protected override void OnLoad(EventArgs e)
{
    MyGanttChart.GanttModel = ganttModel;
    ganttSerializer.SetFileName(ganttModel, filename);
    base.OnLoad(e);
}
```

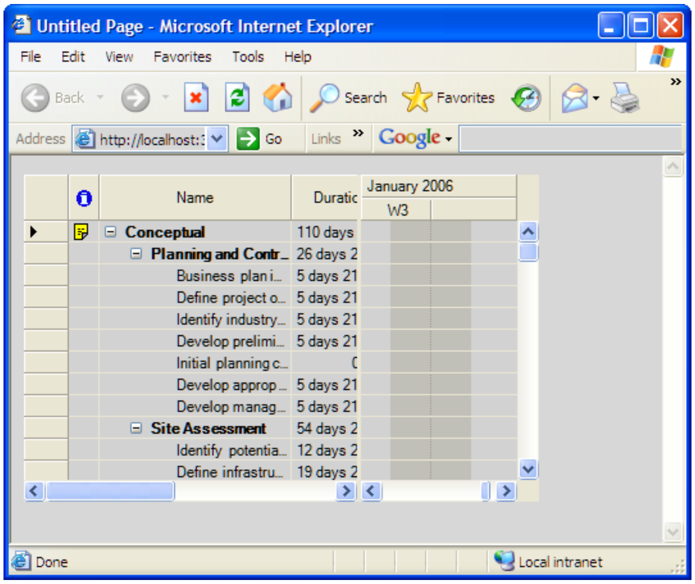
`filename` should be replaced by the location of a SDXL file.

> ***Note:*** *Examples of SDXL files can be found in <install-dir>\Data.*

4. Save all the files.

5. Run the application using the command **Debug** > **Start**.

The Gantt application named **WebGanttApplication** should appear as follows:


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:
Go
Links
Google
January 2006
W3
Done
Local intranet

Name	Duratic
Conceptual	110 days
Planning and Contr…	26 days 2
Business plan i…	5 days 21
Define project o…	5 days 21
Identify industry…	5 days 21
Develop prelimi…	5 days 21
Initial planning c…	(
Develop approp…	5 days 21
Develop manag…	5 days 21
Site Assessment	54 days 2
Identify potentia…	12 days 2
Define infrastru…	19 days 2

Use the scroll bars of the Gantt to navigate in the time line. You can also select rows, or collapse and expand rows by clicking the expand/collapse button of parent rows.

Adding Controls to Interact with the Gantt Chart

This section describes how to add some standard Web Forms controls to interact with the Gantt Chart.

In particular, we'll see how to change the zoom level of the Gantt Chart using buttons.

1. Select a button from the Web Form toolbox tab and drag it onto the web form.
2. From the Properties explorer choose **Appearance** > **Text** and enter the new text: **Zoom In**.
3. Double-click the button to add an event handler. This opens the source code editor. Add a call to the **ZoomIn** method of the **GanttChart** into the button event handler:

   ```
   private void Button1_Click(object sender, System.EventArgs e)
   {
       MyGanttChart.ZoomIn();
   }
   ```

   Select the **Default.aspx** tab to switch back to design view.

1. Select another button from the Web Form toolbox tab and drag it onto the web form.
2. From the Properties explorer choose **Appearance** > **Text** and enter the new text: **Zoom Out**.
3. Double-click the button to add an event handler. This opens the source code editor. Add a call to the **ZoomOut** method of the **GanttChart** into the button event handler:

   ```
   private void Button2_Click(object sender, System.EventArgs e)
   {
       MyGanttChart.ZoomOut();
   }
   ```

   Select the **Default.aspx** tab to switch back to design view.
4. Save the file **Default.aspx**.
5. Run the application using the command **Debug** > **Start**.

   Use the scroll bars of the Gantt Chart and the **Zoom In** and **Zoom Out** buttons to navigate in the time line.

Adding Controls to Modify the Gantt Data Model

This section describes how to add some standard Web Forms controls to modify the Gantt Data Model loaded by the Gantt Chart.

In particular, we'll see how to add an activity to the Gantt Data Model.

1. Select a button from the Web Form toolbox tab and drag it onto the web form.
2. From the Properties explorer choose **Appearance** > **Text** and enter the new text: **New Activity**.
3. Double-click the button to add an event handler. This opens the source code editor. Add the following code into the button event handler:

   ```
   private void Button3_Click(object sender, System.EventArgs e)
   {
       int index = MyGanttChart.NewActivity(
           null,
           "New Activity",
           MyGanttChart.VisibleTimeInterval);
       if (index != -1)
           MyGanttChart.EnsureRowVisible(index);
   }
   ```

   Select the **Default.aspx** tab to switch back to design view.
4. Run the application using the command **Debug** > **Start**.

   Click the **New Activity** button and notice the new added activity.

   Click one more time the **New Activity** button and notice that only one activity has been added. This is because the model is reloaded from the SDXL file at each round-trip. The next step of this tutorial shows how to keep the modifications made to a model between round-trips.

Saving Modifications Across Requests

Each time a client requests the page, a new instance of the page is created to handle the request. This means that everything in the page is re-created including the controls and the data. The ability of ASP.NET to give the illusion that its pages and controls are maintaining some kind of state between different requests relies on the phases the page is put through upon each request. Let's focus onto two phases of the page life cycle: The **LoadViewState** phase, and the **SaveViewState** phase.

The **LoadViewState** phase of a control life cycle initializes the control with data present in the request. This data is called the **ViewState**.

The **SaveViewState** phase of a control life cycle saves the state of the control into the **ViewState**, making it available for the next request.

This section describes how to store the Gantt Data Model into the **ViewState**. Of course, a real application would have used a database instead of storing the model in the **ViewState**, but this goes away from the scope of this tutorial.

1. Change the code of the load event handler to load the SDXL file only the first time the page is requested.

   To do this, double-click in an empty area of the web form. The source code editor opens. Replace the existing code with the following code:

   ```
   protected override void OnLoad(EventArgs e)
   {
       MyGanttChart.GanttModel = ganttModel;
       if (!IsPostBack)
         ganttSerializer.SetFileName(ganttModel, filename);
      base.OnLoad(e);
   }
   ```

   `filename` should be replaced by the location of a SDXL file.

   Note: *Examples of SDXL files can be found in <install-dir>\Data.*

2. Run the application using the command **Debug** > **Start**.

   Click the **New Activity** button and notice that the model now contains only the added activity. This is because the SDXL file is now loaded only during the first page request, and not during postback requests.

3. Switch to code view by right-clicking in the web form and choose **View Code**.

4. Override the **SaveViewState** method by adding the following code:

   ```
   protected override object SaveViewState() {
       System.IO.StringWriter writer = new System.IO.StringWriter();
       ganttSerializer.Serialize(writer, ganttModel);
       ViewState["GanttModelDescription"] = writer.ToString();
       writer.Close();
       return base.SaveViewState();
   }
   ```

   The code above serializes the Gantt Data Model to an XML string, and stores it into the page **ViewState**.

5. Override the **LoadViewState** method by adding the following code:

   ```
   protected override void LoadViewState(object savedState) {
       base.LoadViewState(savedState);
       string xml = ViewState["GanttModelDescription"] as string;
       if (xml != null) {
           System.IO.StringReader reader = new System.IO.StringReader(xml);
           ganttSerializer.Deserialize(reader, ganttModel);
   ```

```
            reader.Close();
        }
    }
```

The code above retrieves the XML description of the scheduling data from the ViewState, and populates the Gantt Data Model.

6. Run the application using the command **Debug** > **Start**.

   Click the **New Activity** button several times, and notice that now the modifications made between several requests are saved.

Synchronizing a Schedule Chart with a Load Chart (Windows Forms Tutorial)

This tutorial walks you through the process of synchronizing a Schedule chart with a Load chart. You will start with the creation of the two charts and then work on the synchronization of their components.

See Also *Creating a Schedule Chart* | *Associating a Gantt Data Model With the Schedule Chart* | *Filling the Schedule Chart With XML Data* | *Adding a Load Chart* | *Synchronizing the Charts* | *Making Reservations Visible at Launch Time*

Creating a Schedule Chart

The Schedule chart is one of the controls that IBM® ILOG® Gantt for .NET provides to display scheduling information. The Schedule chart is divided in three parts:

- The left part is a resource table (an instance of the class GanttTable) which displays resource information from the data model.
- The right part is a Gantt Sheet (an instance of the class GanttSheet) which displays the resource reservations.
- Just above the Gantt sheet appears a zoomable time scale (an instance of the class TimeScale).

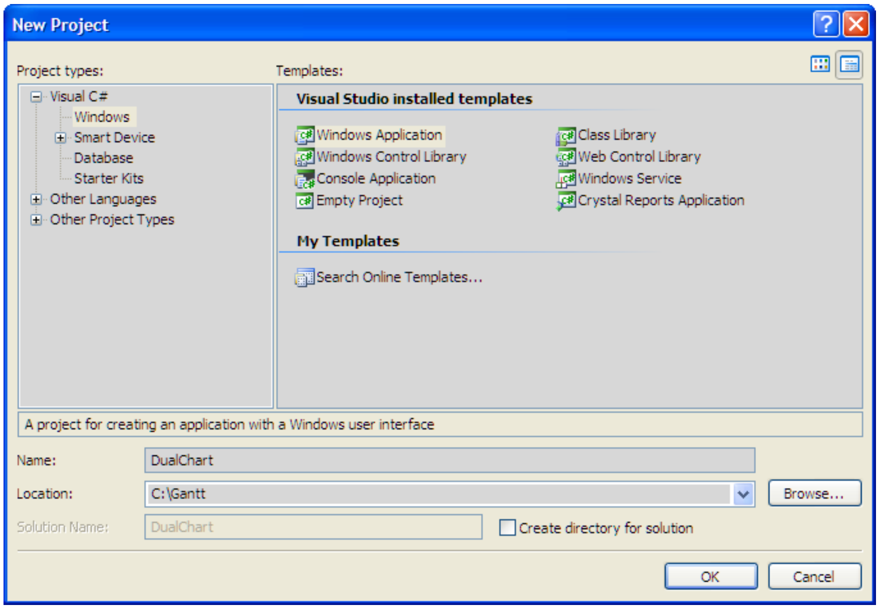
A standard, adjustable splitter separates the left part from the right part.

To create a Schedule chart, perform the following steps:

1. Create a project of type Windows® Forms.

   On the **File** menu, click **New**, and then click **Project**.

   The New Project window appears.



New Project

Project types:
- Visual C#
  - Windows
  - Smart Device
  - Database
  - Starter Kits
- Other Languages
- Other Project Types

Templates:

Visual Studio installed templates

Windows Application, Class Library, Windows Control Library, Web Control Library, Console Application, Windows Service, Empty Project, Crystal Reports Application

My Templates

Search Online Templates...

A project for creating an application with a Windows user interface

Name: DualChart

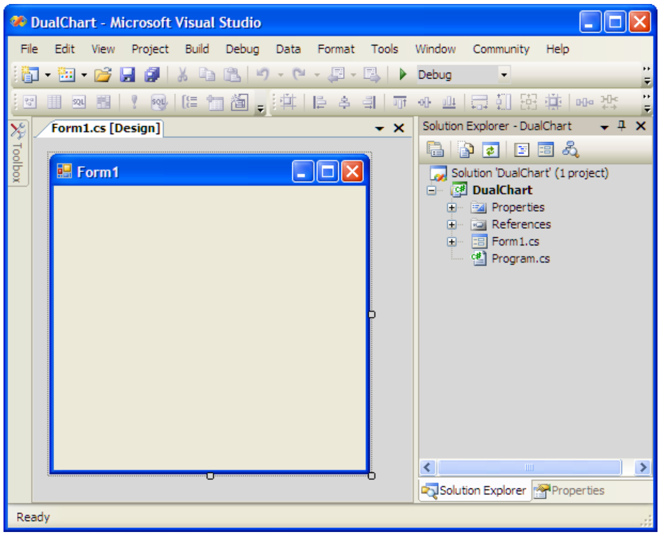
Location: C:\Gantt

Solution Name: DualChart

Create directory for solution

2. In the left window, click **Visual C# > Windows**.
3. In the right window, click **Windows Application**.
4. Name the project DualChart.
5. Place the project in **C:\Gantt**.

   Visual Studio®.NET creates a Form1 and stores it by default in a file named **Form1.cs**.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]
Form1
Solution Explorer - DualChart
Solution 'DualChart' (1 project)
DualChart
Properties
References
Form1.cs
Program.cs
Solution Explorer
Properties
Ready

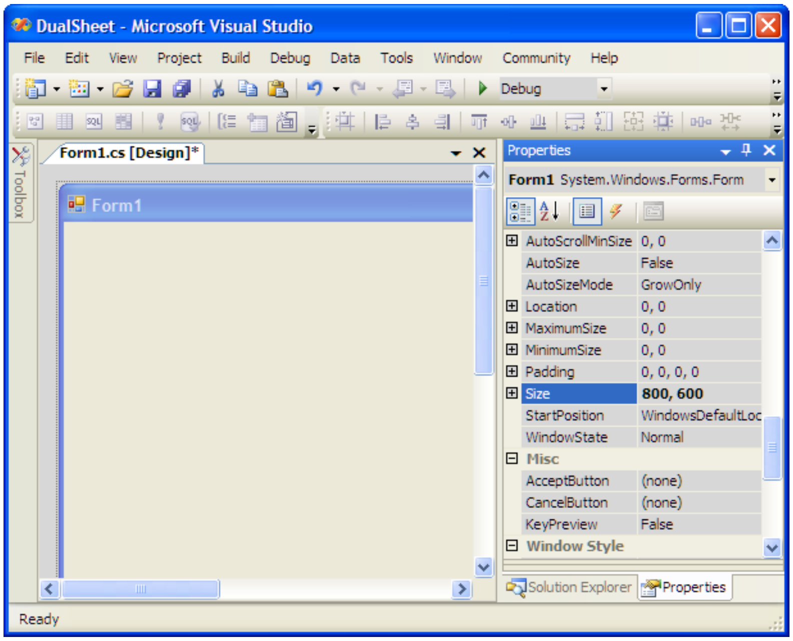
6. Resize the window.

Click Form1 and select the Properties tab at the bottom right of the window.

From the Properties explorer choose **Layout** > **Size**. Expand the property and enter the following values:

Width: 800

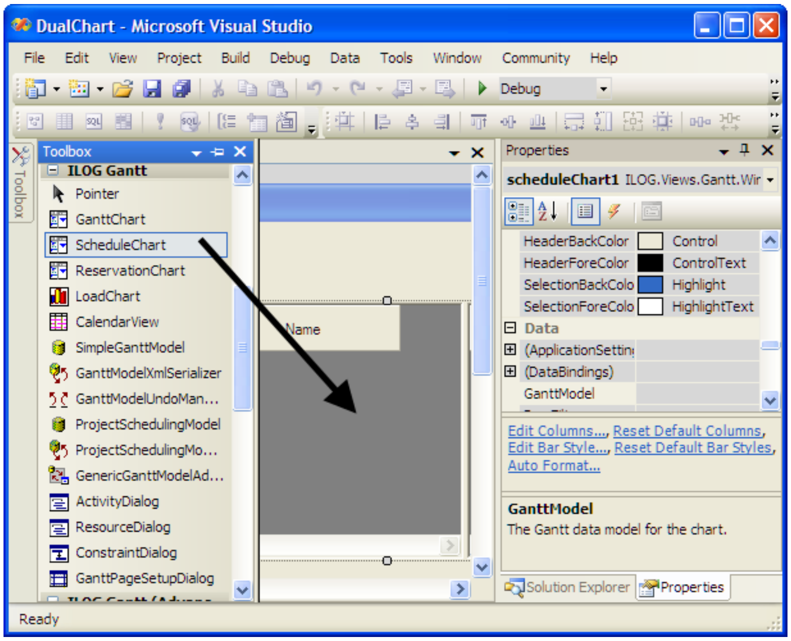
Height: 600


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Properties
Solution Explorer
Properties
Ready

Form1 System.Windows.Forms.Form

AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
Location	0, 0
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	800, 600
StartPosition	WindowsDefaultLoc
WindowState	Normal
Misc	
AcceptButton	(none)
CancelButton	(none)
KeyPreview	False
Window Style	

7. From the IBM ILOG Gantt for .NET toolbox select a **ScheduleChart** object and drag it to Form1.



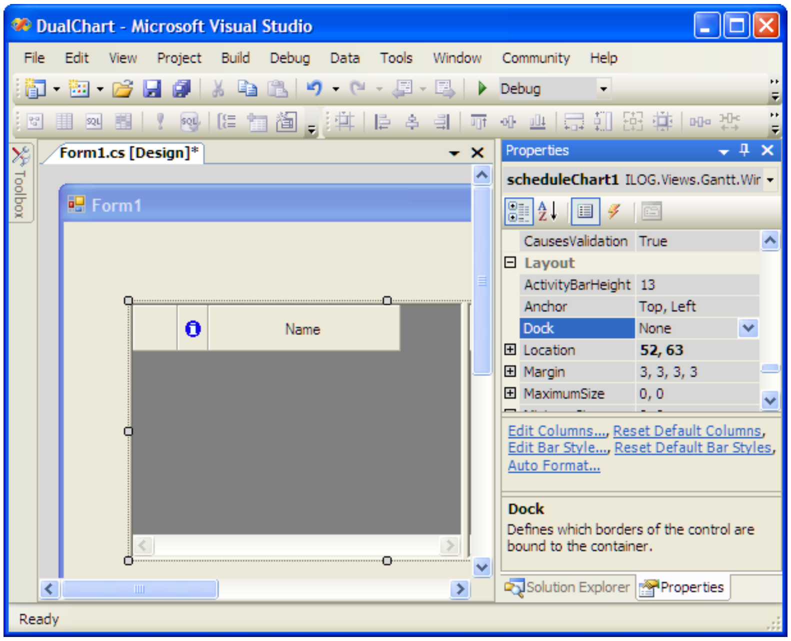
scheduleChart1 ILOG.Views.Gantt.Wir

HeaderBackColor	Control
HeaderForeColor	ControlText
SelectionBackColo	Highlight
SelectionForeColo	HighlightText
Data	
(ApplicationSettin	
(DataBindings)	
GanttModel	

GanttModel
The Gantt data model for the chart.

8. Change the Dock property to make the schedule chart occupy the entire form.

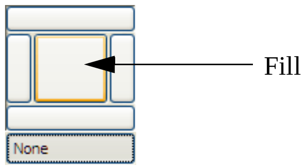
Click the schedule chart in Form1and from the Properties explorer choose **Layout** > **Dock**.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Form1
Name
Properties
scheduleChart1 ILOG.Views.Gantt.Wir
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

CausesValidation	True
Layout	
ActivityBarHeight	13
Anchor	Top, Left
Dock	None
Location	52, 63
Margin	3, 3, 3, 3
MaximumSize	0, 0

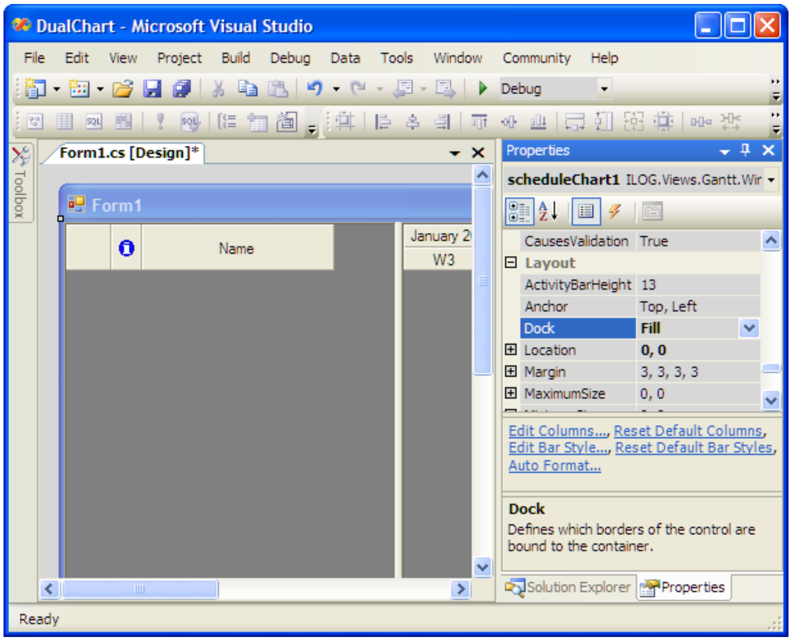
Dock
Defines which borders of the control are bound to the container.

9. Clicking the arrow next to the Dock property opens the following box:



10. Select the area corresponding to the Fill option.

The schedule chart now occupies the entire form and appears as illustrated below:

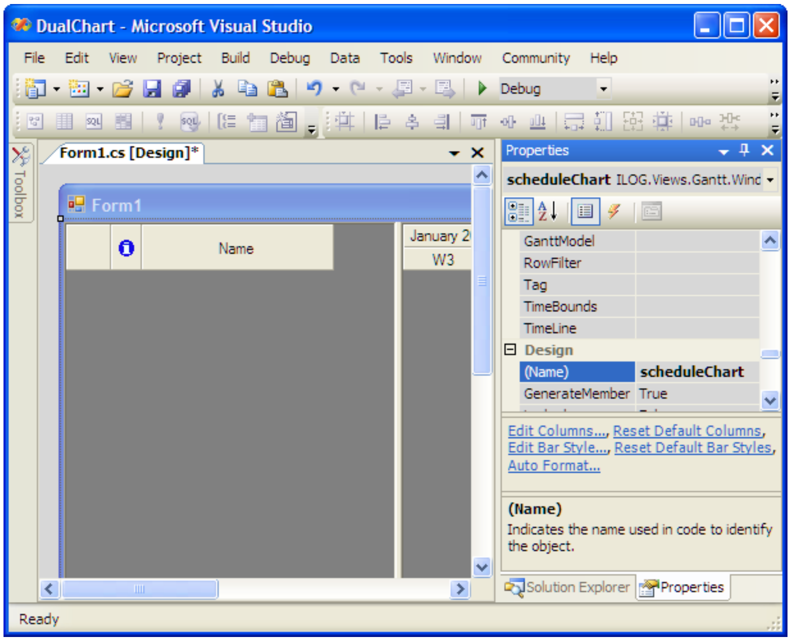

DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Name
January 2
W3
Properties
scheduleChart1 ILOG.Views.Gantt.Wir
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

CausesValidation	True
Layout	
ActivityBarHeight	13
Anchor	Top, Left
Dock	Fill
Location	0, 0
Margin	3, 3, 3, 3
MaximumSize	0, 0

Dock
Defines which borders of the control are bound to the container.

11. Rename the schedule chart.

Click the schedule chart in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **scheduleChart**.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Form1
Name
January 2
W3
Toolbox
Properties
scheduleChart ILOG.Views.Gantt.Winc
Solution Explorer
Properties
Ready

GanttModel	
RowFilter	
Tag	
TimeBounds	
TimeLine	
Design	
(Name)	scheduleChart
GenerateMember	True

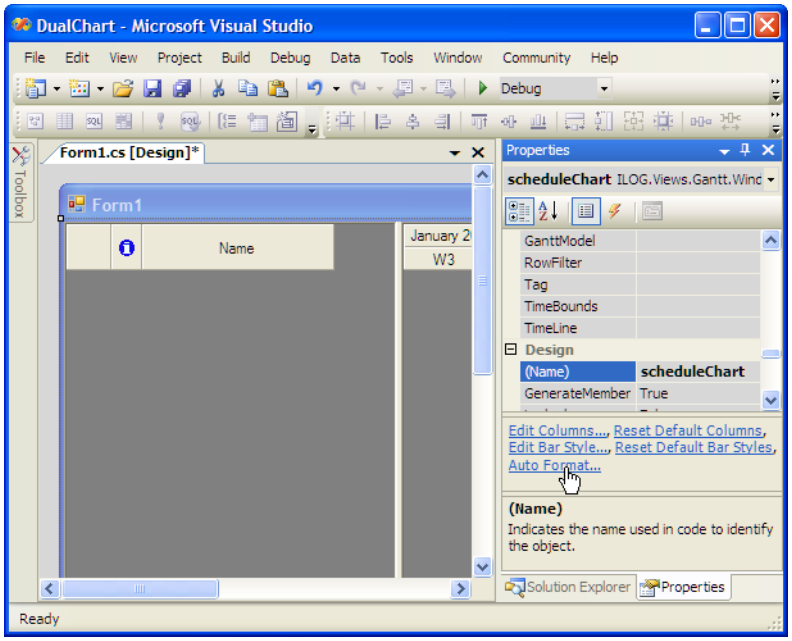
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...

(Name)
Indicates the name used in code to identify the object.

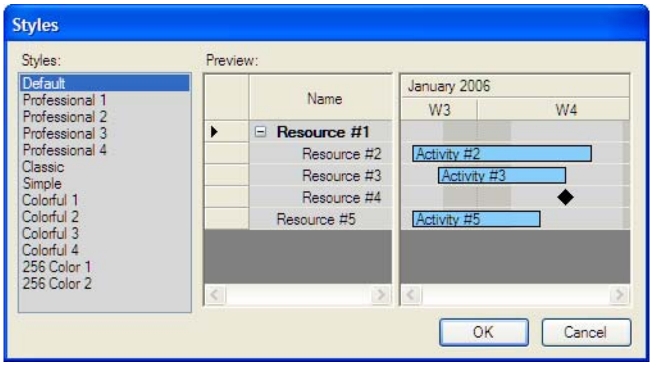
The appearance of the schedule chart can be completely customized to fit your needs.

12. Define a style for the schedule chart you have just created.

You can either right-click the schedule chart and select AutoFormat in the contextual menu, or click AutoFormat at the bottom of the Properties explorer.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Form1
Name
January 2
W3
Toolbox
Properties
scheduleChart ILOG.Views.Gantt.Wind
GanttModel
RowFilter
Tag
TimeBounds
TimeLine
Design
(Name) scheduleChart
GenerateMember True
Edit Columns..., Reset Default Columns,
Edit Bar Style..., Reset Default Bar Styles,
Auto Format...
(Name)
Indicates the name used in code to identify the object.
Solution Explorer
Properties
Ready

The Styles window appears:



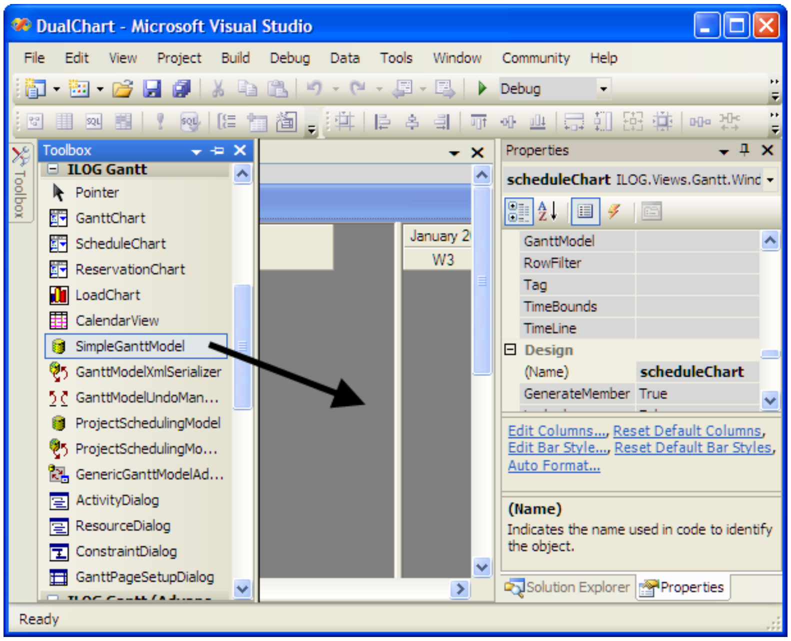
Select a style and click OK. The style that you selected is automatically applied to the schedule chart.

13. Save the file **Form1.cs**.

Associating a Gantt Data Model With the Schedule Chart

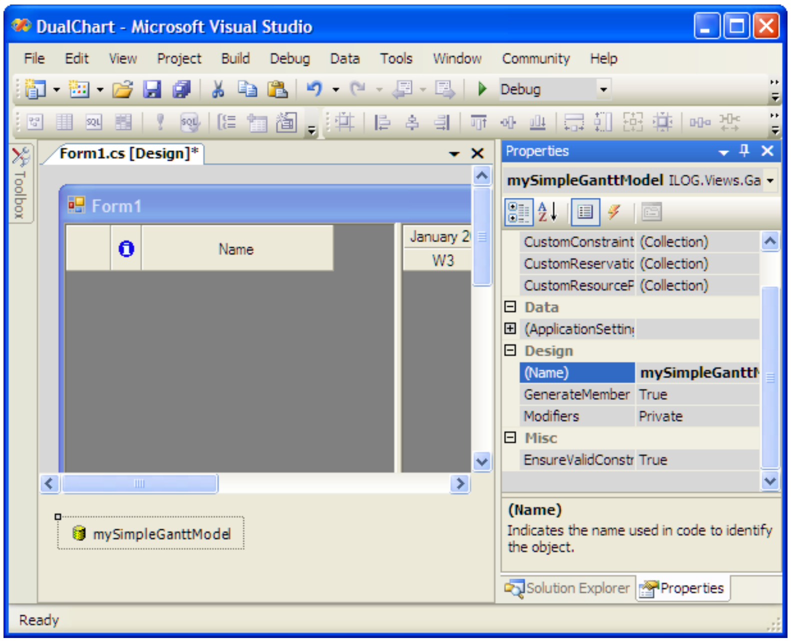
The data model contains the scheduling information you want to display in the schedule chart. In the following steps, you will learn how to bind a simple Gantt data model to the schedule chart.

1. From the IBM® ILOG® Gantt for .NET toolbox select the **SimpleGanttModel** object and drag it to Form1.



2. Rename the **simpleGanttModel** object.

   Click the **SimpleGanttModel1** object in Form1. From the Properties explorer choose **Design** > **Name** and enter the following name: **mySimpleGanttModel.**


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Name
January 2
W3
mySimpleGanttModel
Properties
mySimpleGanttModel ILOG.Views.Ga
Solution Explorer
Properties
Ready

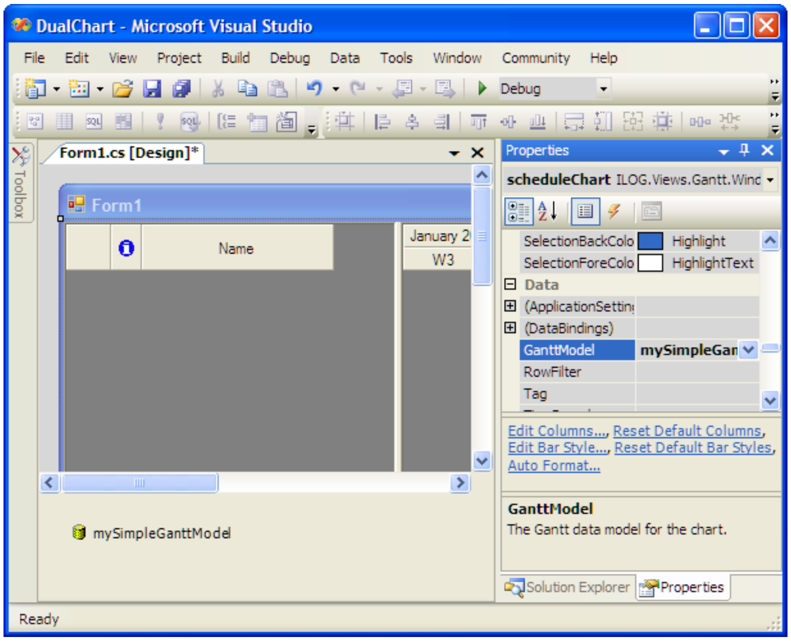
CustomConstraint	(Collection)
CustomReservatic	(Collection)
CustomResourceP	(Collection)
⊟ Data	
⊞ (ApplicationSettin	
⊟ Design	
(Name)	mySimpleGanttM
GenerateMember	True
Modifiers	Private
⊟ Misc	
EnsureValidConstr	True

(Name)
Indicates the name used in code to identify the object.

3. Associate **mySimpleGanttModel** with **scheduleChart**.

   Click the schedule chart in Form1 and from the Properties explorer select **Data > GanttModel**.

   Click the arrow of the drop-down list next to the GanttModel property and choose **mySimpleGanttModel**.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Name
January 2
W3
mySimpleGanttModel
Properties
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

scheduleChart ILOG.Views.Gantt.Winc

SelectionBackColo	Highlight
SelectionForeColo	HighlightText
Data	
(ApplicationSettin	
(DataBindings)	
GanttModel	mySimpleGan
RowFilter	
Tag	

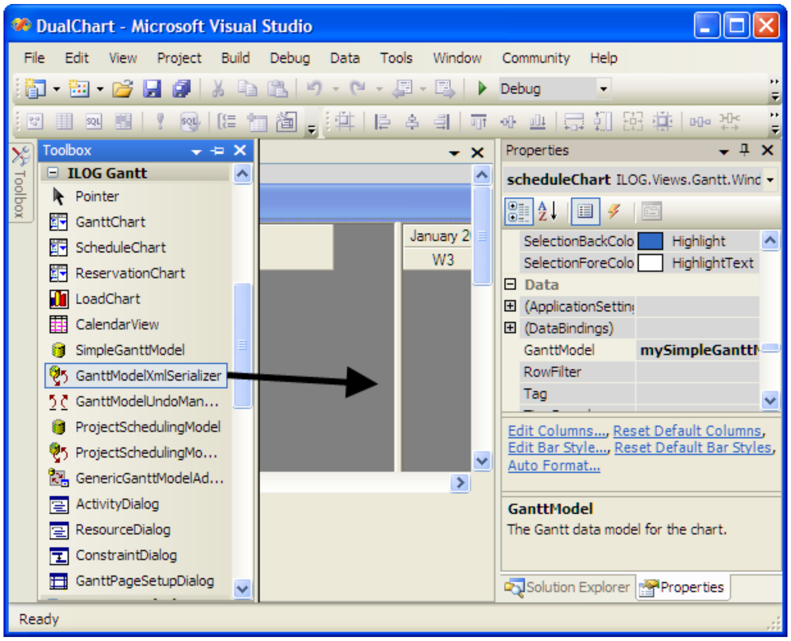
GanttModel
The Gantt data model for the chart.

Filling the Schedule Chart With XML Data

IBM® ILOG® Gantt for .NET allows you to serialize and deserialize scheduling data to and from Scheduling Data Exchange Language (SDXL) files. This is accomplished through the **GanttModelXmlSerializer** class.

To deserialize scheduling data from SDXL files and use this information to fill the schedule chart, perform the following steps.

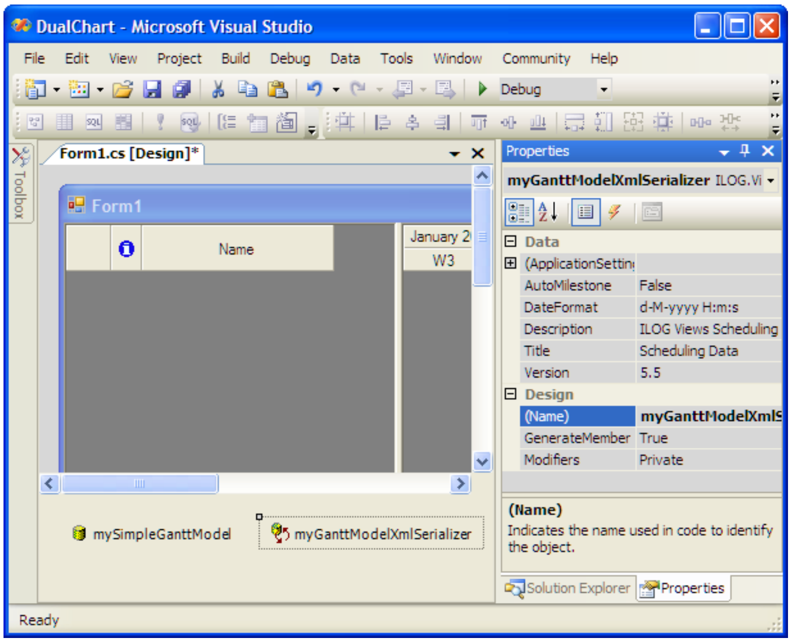
1. From the IBM ILOG Gantt for .NET toolbox select the **GanttModelXmlSerializer** object and drag it to Form1.



2. Rename the Gantt model XML serializer.

Click the **ganttModelXmlSerializer1** object in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **myGanttModelXmlSerializer**.

3. Click **mySimpleGanttModel**. You will notice that a new property named **FileName on myGanttModelXmlSerializer** has been created.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Name
January 2
W3
mySimpleGanttModel
myGanttModelXmlSerializer
Properties
Solution Explorer
Properties
Ready

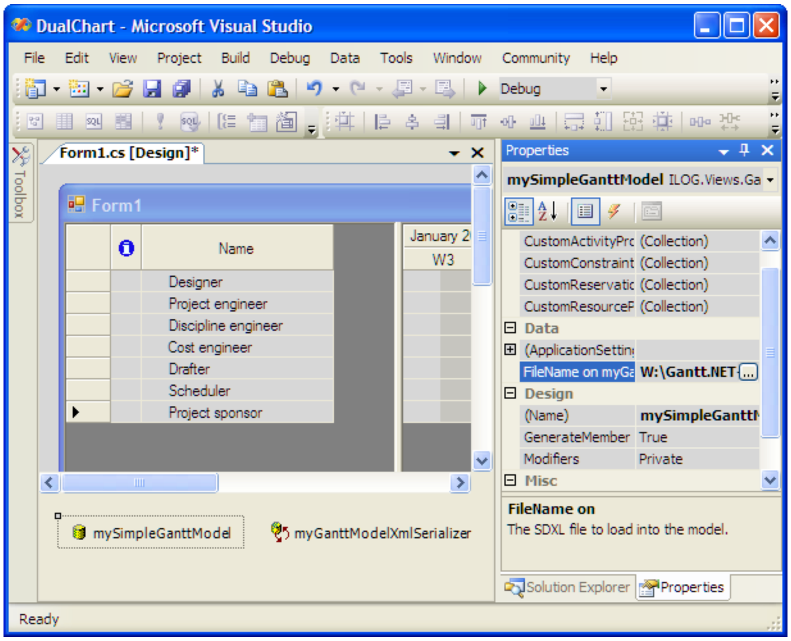
myGanttModelXmlSerializer ILOG.Vi

Data	
(ApplicationSettin	
AutoMilestone	False
DateFormat	d-M-yyyy H:m:s
Description	ILOG Views Scheduling
Title	Scheduling Data
Version	5.5
Design	
(Name)	myGanttModelXmlS
GenerateMember	True
Modifiers	Private

(Name)
Indicates the name used in code to identify the object.

4. Select the property **FileName on ganttModelXmlSerializer** and click the three-dots button next to it. Choose an SDXL file to fill the schedule chart.

__Note:__ Examples of SDXL files can be found in <install-dir>\Data.



CustomActivityPro	(Collection)
CustomConstraint	(Collection)
CustomReservatio	(Collection)
CustomResourceP	(Collection)
Data	
(ApplicationSettin	
FileName on myGa	W:\Gantt.NET
Design	
(Name)	mySimpleGanttM
GenerateMember	True
Modifiers	Private
Misc	

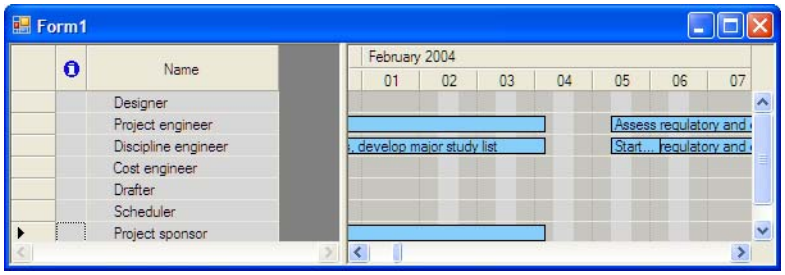
FileName on
The SDXL file to load into the model.

Data contained in the **Engineering.sdxl** file are displayed in the schedule chart.

5. Save the file **Form1.cs**.

6. Run the application using the command **Debug** > **Start**.

The application will appear as follows:



7. Close the application you have just executed and go back to Form1.cs.

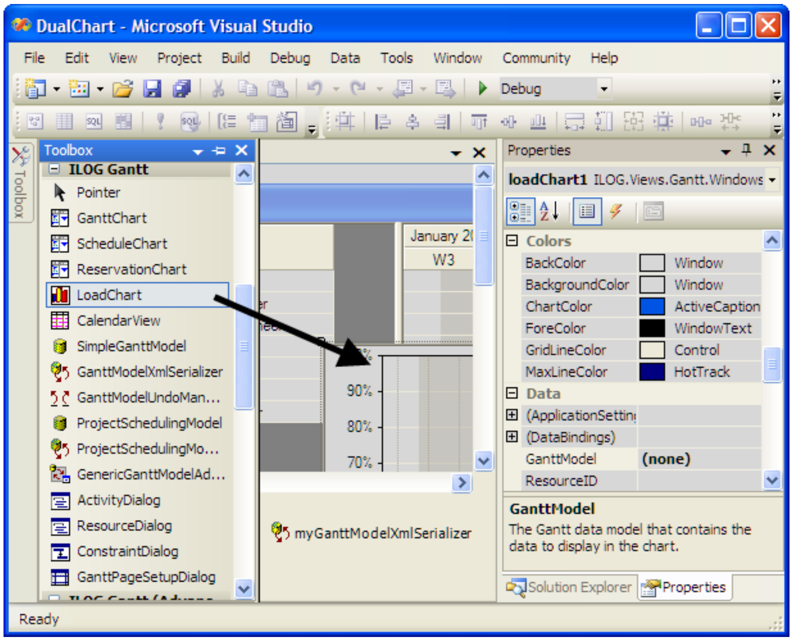
Adding a Load Chart

Now you will learn how to create a Load chart and associate it with the Schedule chart.

The Load chart is the graphical representation of the load of a resource over the time. It is composed of a chart area where the x-axis represents the time and the y-axis represents the load of the resource.

To create the Load chart, perform the following steps:

1. From the IBM® ILOG® Gantt for .NET toolbox select the **LoadChart** object and drag it to Form1.

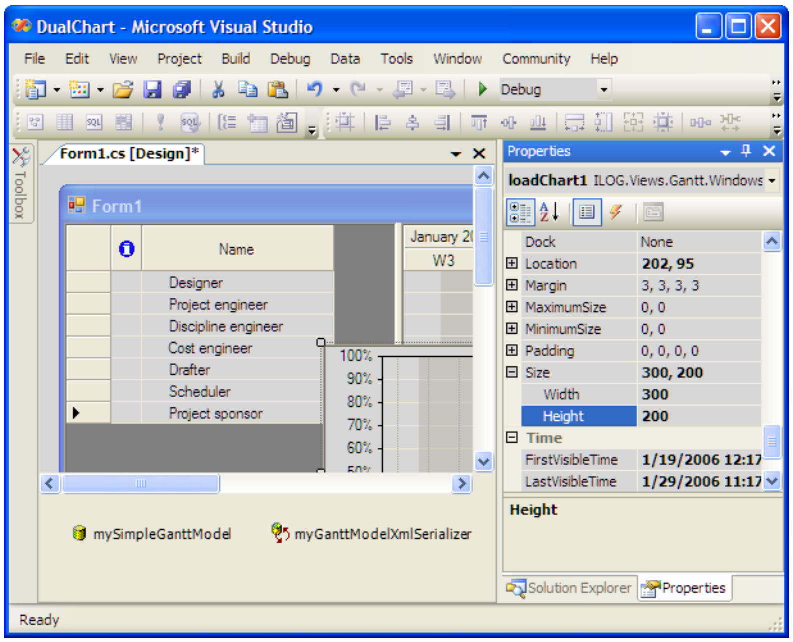


2. Rename the Load chart.

   Click the Load chart in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **loadChart**.

3. Change the height of the Load chart.

   From the Properties explorer choose **Layout** > **Size** > **Height** and enter **200**.

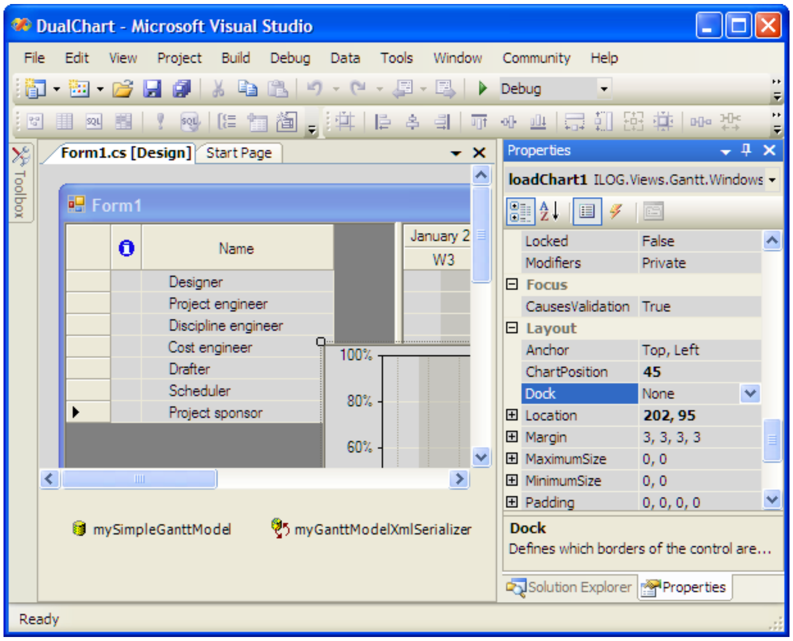


loadChart1 ILOG.Views.Gantt.Windows

Dock	None
⊞ Location	202, 95
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
⊞ Padding	0, 0, 0, 0
⊟ Size	300, 200
Width	300
Height	200
⊟ Time	
FirstVisibleTime	1/19/2006 12:17
LastVisibleTime	1/29/2006 11:17

Height

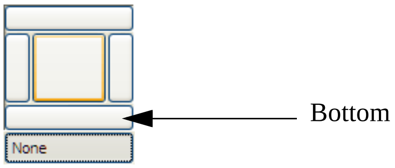
4. Change the Dock property to make the Load chart occupy the bottom of the Form. Click the Load chart and from the Properties explorer choose **Layout** > **Dock**.



Locked	False
Modifiers	Private
⊟ Focus	
CausesValidation	True
⊟ Layout	
Anchor	Top, Left
ChartPosition	45
Dock	None
⊞ Location	202, 95
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
⊞ Padding	0, 0, 0, 0

Dock
Defines which borders of the control are...

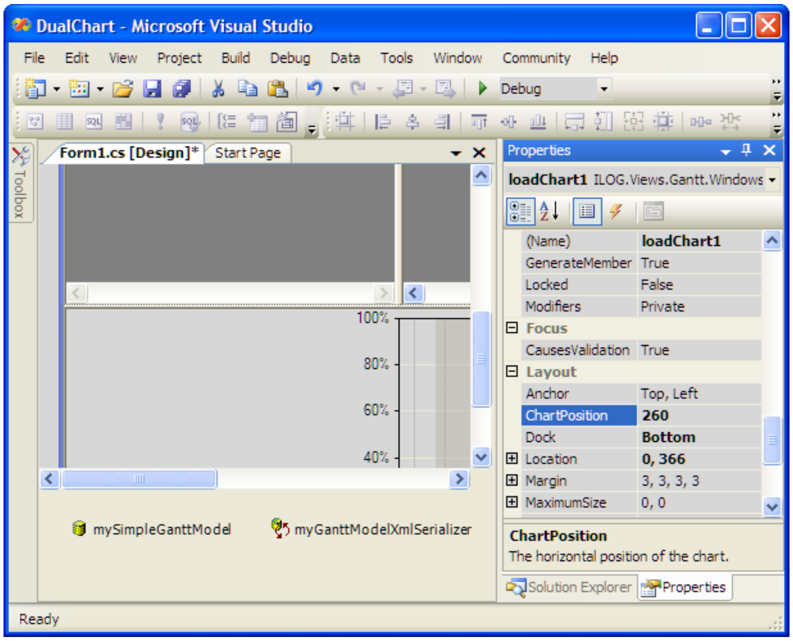
Clicking the arrow next to the Dock property shows the following control:



5. Select the area corresponding to the Bottom option.

6. Right-click the schedule chart and select Bring to Front.

7. Align the Load chart with the splitter of the Schedule chart.
Click the Load chart in Form1. From the Properties explorer choose **Layout** > **ChartPosition** and enter **260**.

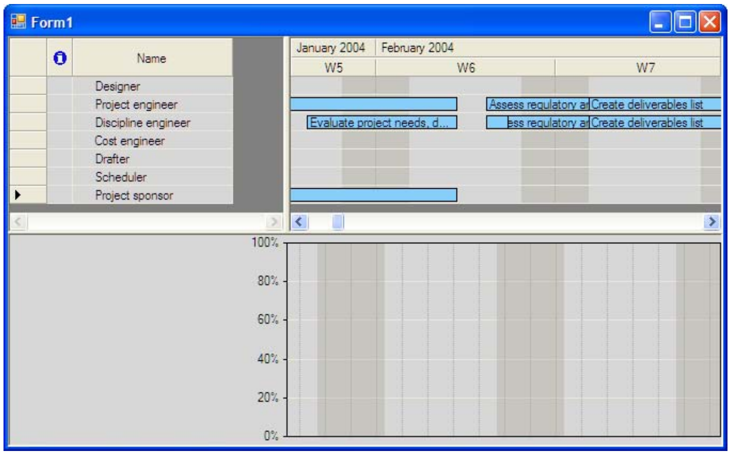


loadChart1 ILOG.Views.Gantt.Windows

(Name)	loadChart1
GenerateMember	True
Locked	False
Modifiers	Private
⊟ Focus	
CausesValidation	True
⊟ Layout	
Anchor	Top, Left
ChartPosition	260
Dock	Bottom
⊞ Location	0, 366
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0

ChartPosition
The horizontal position of the chart.

Now that you have added the Load chart, the application appears as follows:



8. Close the application you have just executed and go back to Form1.cs.

Synchronizing the Charts

Now that you have created the Schedule chart and the Load chart, you will learn how to synchronize them.

See Also *Synchronizing the Visible Time Range* | *Synchronizing the Grids* | *Synchronizing the Charts Splitter* | *Synchronizing the Resource Selection*

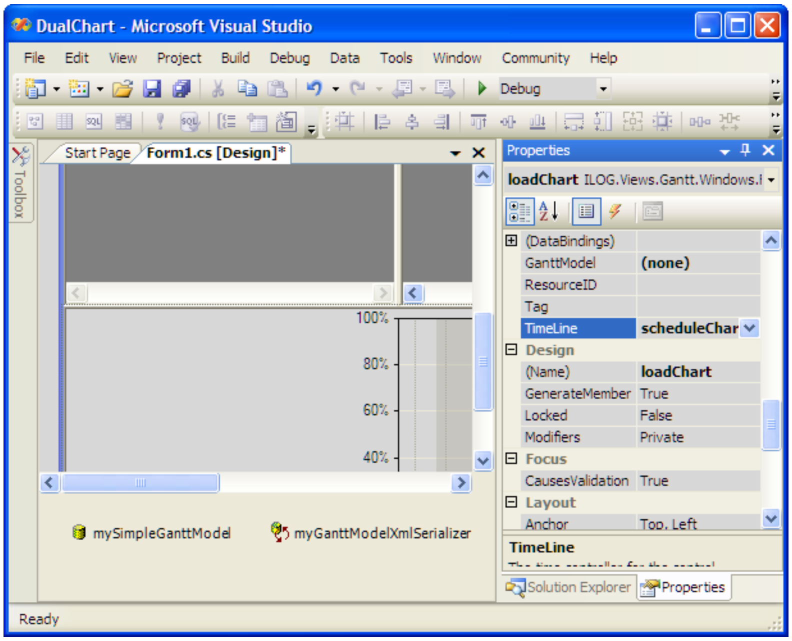
Synchronizing the Visible Time Range

The display of the time information is controlled by an object called *time line* of the **TimeControl**. This object defines the first visible time of the **TimeControl** instance as well as the way to convert time information to pixels on the screen.

The Schedule chart and Load chart are synchronized when they share the same *time line,* so that any modification of the visible time of the Schedule chart applies simultaneously to the Load chart.

To synchronize the visible time range, perform the following steps:

1. Select the Load chart.
2. From the Properties explorer choose **TimeLine** and from the drop-down list select **scheduleChart**.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Start Page
Form1.cs [Design]*
100%
80%
60%
40%
mySimpleGanttModel
myGanttModelXmlSerializer
Properties
Solution Explorer
Properties
Ready

loadChart ILOG.Views.Gantt.Windows.I

(DataBindings)	
GanttModel	(none)
ResourceID	
Tag	
TimeLine	scheduleChar
Design	
(Name)	loadChart
GenerateMember	True
Locked	False
Modifiers	Private
Focus	
CausesValidation	True
Layout	
Anchor	Top, Left

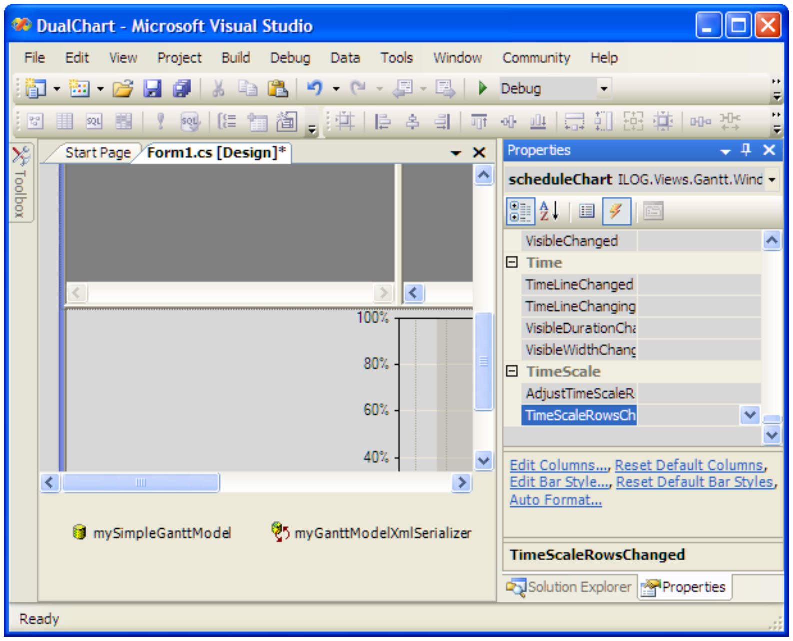
TimeLine

Synchronizing the Grids

When the Schedule chart is used in sync with another control displaying time information, for example the Load chart, it is interesting to synchronize the vertical grid lines of the control (the Time grid) with the ticks of the Time Scale.

To synchronize the grids, perform the following steps:

1. Select the Schedule chart.
2. From the Properties explorer select the Events page by clicking the ⚡ icon.
3. Double-click the **TimeScale** > **TimeScaleRowsChanged** event.


DualChart - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Start Page
Form1.cs [Design]*
100%
80%
60%
40%
mySimpleGanttModel
myGanttModelXmlSerializer
Properties
scheduleChart ILOG.Views.Gantt.Winc
VisibleChanged
Time
TimeLineChanged
TimeLineChanging
VisibleDurationCha
VisibleWidthChang
TimeScale
AdjustTimeScaleR
TimeScaleRowsCh
Edit Columns..., Reset Default Columns, Edit Bar Style..., Reset Default Bar Styles, Auto Format...
TimeScaleRowsChanged
Solution Explorer
Properties
Ready

The code editor opens with the cursor placed automatically at the appropriate location.

4. Insert the following code:

```
loadChart.TimeGrids.Synchronize(scheduleChart.TimeGrids);
```

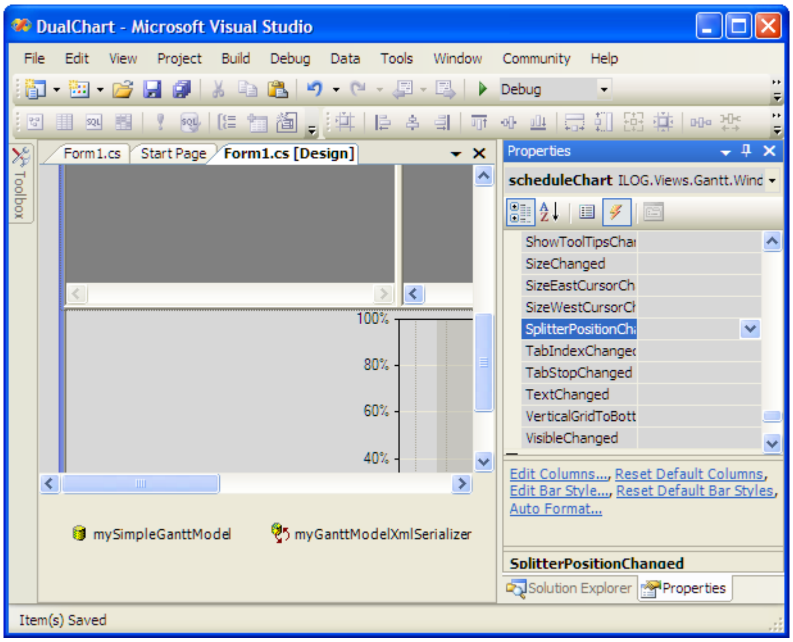
5. Save the file **Form1.cs** and go back to Design mode by selecting **View** > **Designer** on the menu bar.

Synchronizing the Charts Splitter

The Schedule chart contains a splitter that separates the Gantt table from the Gantt sheet. A double click on the splitter will make the table fully visible. The splitter can be controlled by means of the following properties: **SplitterPosition** and **SplitterWidth**.

To synchronize the position of the splitter between the Schedule chart and Load chart, perform the following steps:

1. Select the Schedule chart.
2. Got to the Properties explorer and select the Events page by clicking the icon.
3. Double-click the **SplitterPositionChanged** event.



The code editor opens with the cursor automatically placed at the appropriate location.

4. Insert the following code:

```
loadChart.ChartPosition = scheduleChart.SplitterPosition +
scheduleChart.SplitterWidth;
```

5. Save the file **Form1.cs** and go back to Design mode by selecting **View** > **Designer** on the menu bar.

Synchronizing the Resource Selection

The resource selection is synchronized between the Schedule chart and Load chart when the resource, once selected, is displayed in both charts.

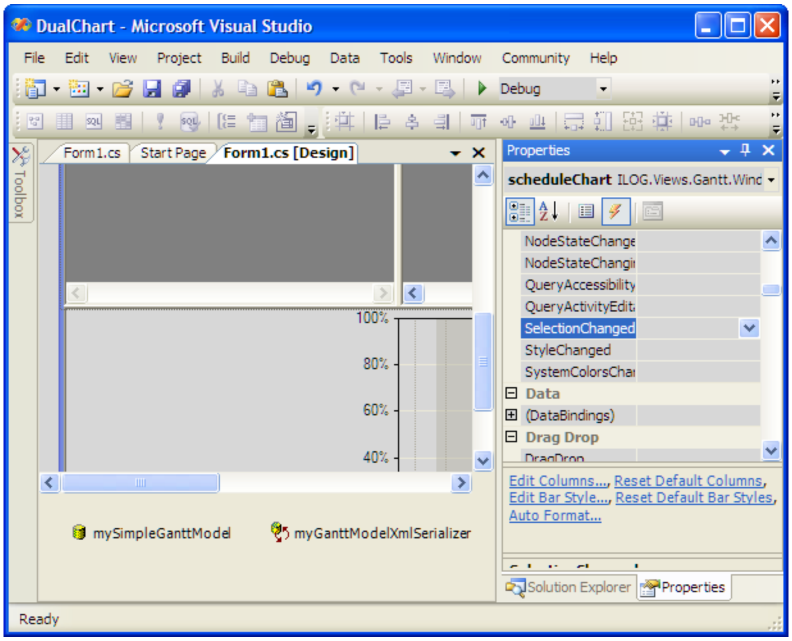
To obtain this, the Load chart must be connected to a resource using its Resource property. Once connected to a resource, the Load chart listens to events in the Gantt data model and updates himself when the resource changes or when the reservation changes.

To synchronize the resource selection, perform the following steps:

1. Select the Schedule chart.

2. From the Properties explorer select the Events page by clicking the ⚡ icon.

3. Double-click the **Behavior** > **SelectionChanged** event.



The code editor opens with the cursor placed automatically at the appropriate location.

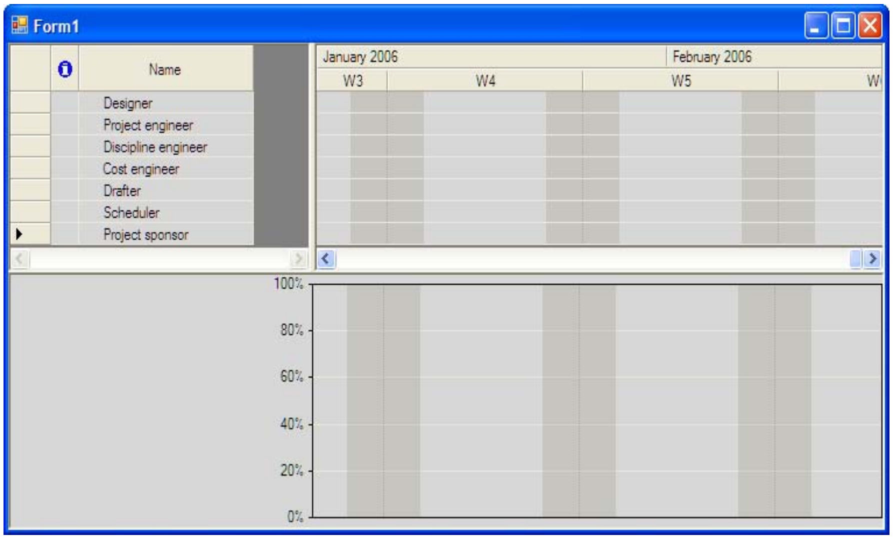
4. Insert the following code:

```
ILOG.Views.Gantt.Data.IResource[] resources =
scheduleChart.GetSelectedResources();
if (resources.Length == 1)
   loadChart.Resource = resources[0];
else
    loadChart.Resource = null;
```

5. Save the file **Form1.cs**.

6. Run the application using the command **Debug** > **Start**.

The application should appear as follows:



7. Close the application you have just executed.

Making Reservations Visible at Launch Time

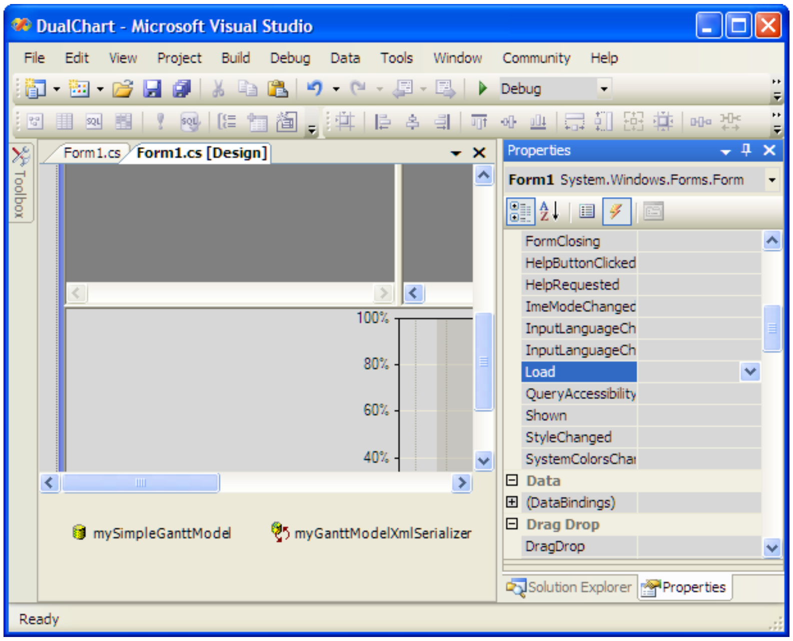
When the application is launched, rows and columns appear empty. The scheduling information is there but you have to scroll to view it.

In the following steps you will learn how to make the reservations immediately visible in the chart when the application is launched.

1. Select Form1.

2. From the Properties explorer select the Events page by clicking the ⚡ icon.

3. Double-click the **Behavior** > **Load** event.



The code editor opens with the cursor placed automatically at the appropriate location.

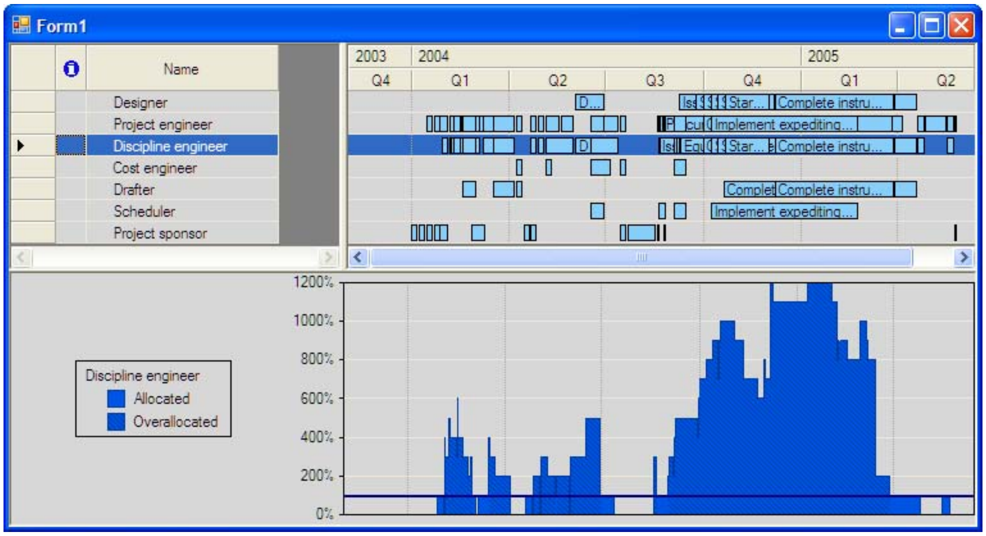
4. Insert the following code:

```
scheduleChart.GanttSheet.ZoomToFit(false);
```

5. Save the file **Form1.cs**.

6. Run the application using the command **Debug** > **Start**.

The application should appear as follows:



Creating a Custom Gantt Representation (Windows Forms Tutorial)

This tutorial will walk you through the process of creating a custom Gantt chart representation made of an activity table and two scroll activity sheets. You will start with the creation of the table and then add one by one the scroll activity sheets.

See Also *Creating the Activity Table* | *Associating a Gantt Model With the Activity Table* | *Filling the Activity Table With XML Data* | *Adding the Scroll Activity Sheets* | *Making Reservation Visible at Launch Time*

Creating the Activity Table

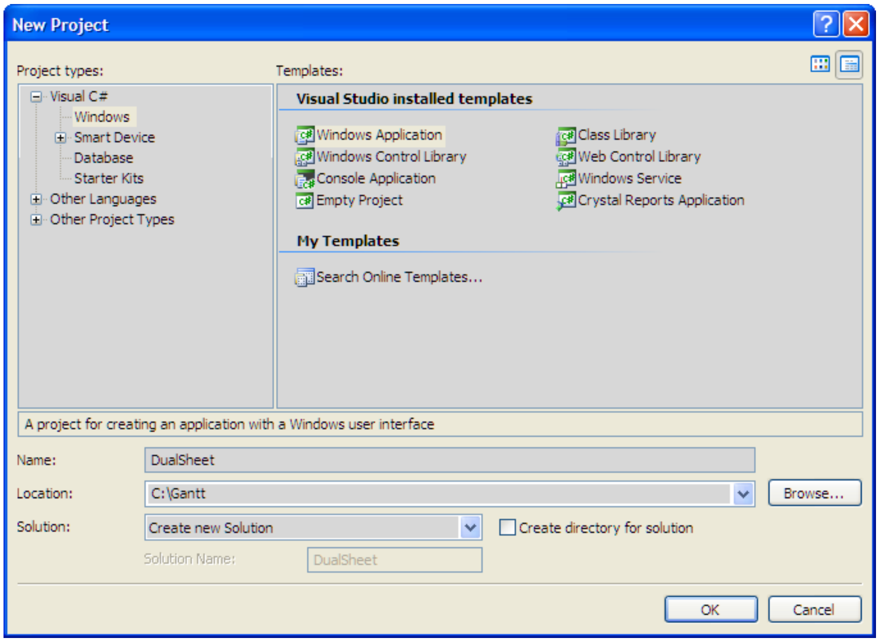
The Activity table is one of the controls that IBM® ILOG® Gantt for .NET provides to display scheduling information. The custom representation you are going to create is composed of three main parts: an activity table (instance of the class GanttTable) on the left part displays information from the data model; two scroll activity sheets (instances of the class GanttSheet) on the right part display the activity reservations.

To create an Activity table, perform the following steps:

1. Create a project of type Windows® Forms.

   On the **File** menu, click **New**, and then click **Project**.

The New Project window appears.



A project for creating an application with a Windows user interface

Name: DualSheet

Location: C:\Gantt

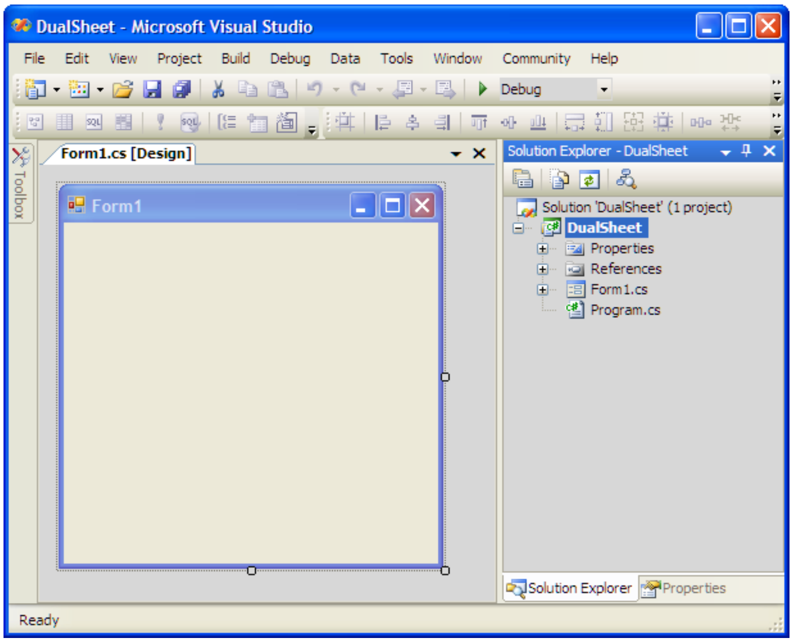
Solution: Create new Solution

Create directory for solution

Solution Name: DualSheet

2. In the left window, click **Visual C#** > **Windows**.
3. In the right window, click **Windows Application**.
4. Name the project DualSheet.
5. Place the project in **C:\Gantt**.

   Visual Studio®.NET creates a form named Form1 and stores it by default in the file Form1.cs.


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]
Form1
Solution Explorer - DualSheet
Solution 'DualSheet' (1 project)
DualSheet
Properties
References
Form1.cs
Program.cs
Solution Explorer
Properties
Ready

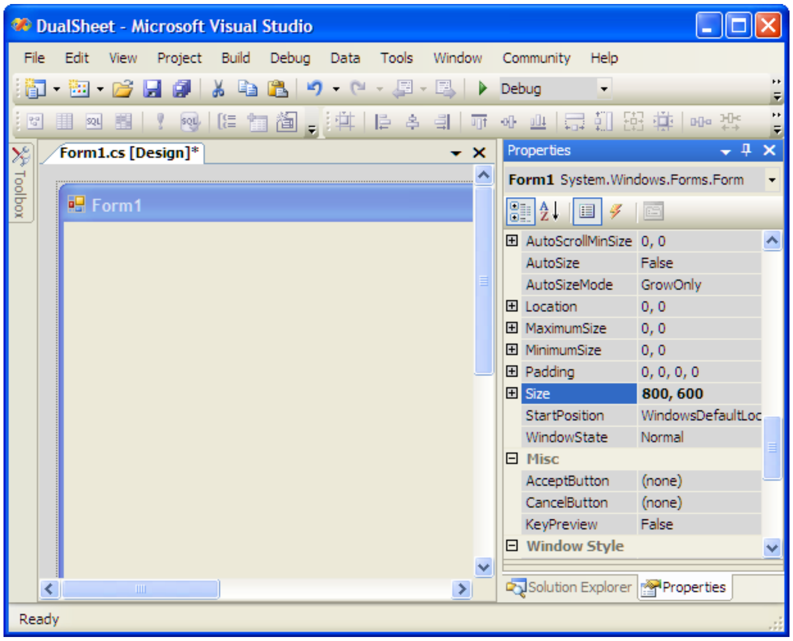
6. Resize the window.

Click Form1 and select the Properties tab at the bottom right of the window.

From the Properties explorer choose **Layout** > **Size.** Expand the property and enter the following values:

Width: 800

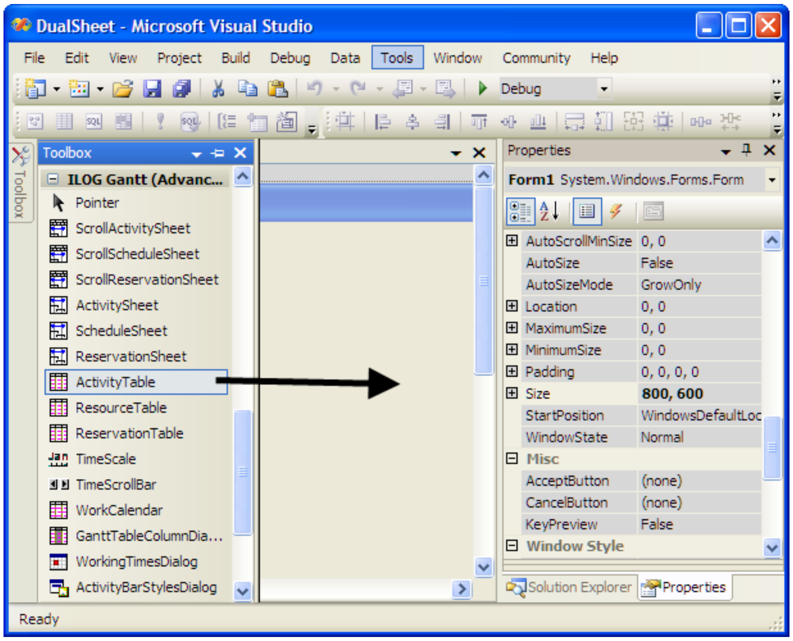
Height: 600


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Properties
Solution Explorer
Properties
Ready

Form1 System.Windows.Forms.Form

AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
Location	0, 0
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	**800, 600**
StartPosition	WindowsDefaultLoc
WindowState	Normal
Misc	
AcceptButton	(none)
CancelButton	(none)
KeyPreview	False
Window Style	

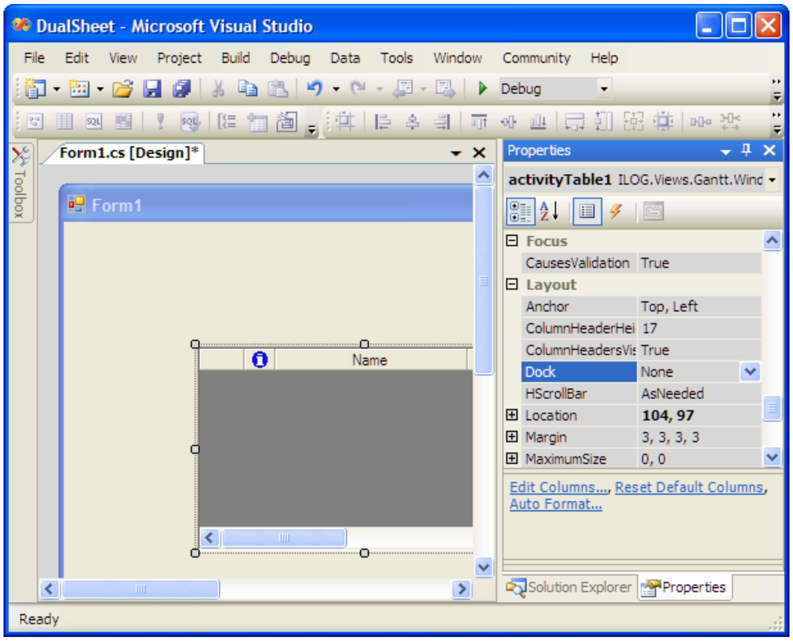
7. From the IBM ILOG Gantt for .NET toolbox select the **ActivityTable** object and drag it to Form1.



AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
Location	0, 0
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	**800, 600**
StartPosition	WindowsDefaultLoc
WindowState	Normal
Misc	
AcceptButton	(none)
CancelButton	(none)
KeyPreview	False
Window Style	

8. Change the Dock property to make the Activity Table occupy the entire form.

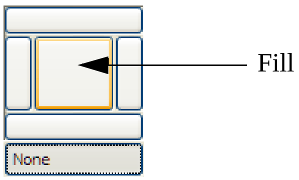
Click the Activity Table and from the Properties explorer choose **Layout** > **Dock**.


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Form1
Name
Toolbox
Properties
activityTable1 ILOG.Views.Gantt.Wind
Solution Explorer
Properties
Ready

Focus	
CausesValidation	True
Layout	
Anchor	Top, Left
ColumnHeaderHei	17
ColumnHeadersVis	True
Dock	None
HScrollBar	AsNeeded
Location	104, 97
Margin	3, 3, 3, 3
MaximumSize	0, 0

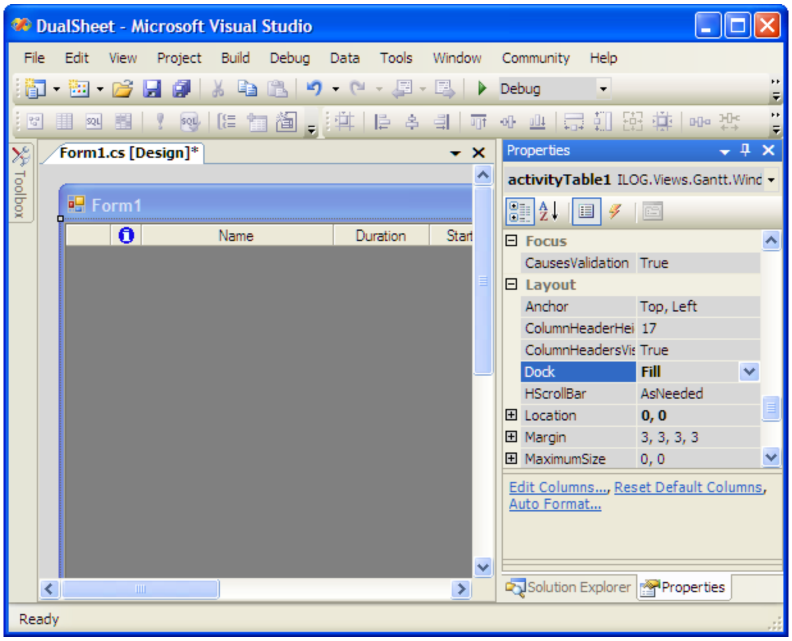
Edit Columns..., Reset Default Columns, Auto Format...

Clicking the arrow next to the Dock property shows the following control:



9. Select the area corresponding to the Fill option.

   Form1 now looks like this:

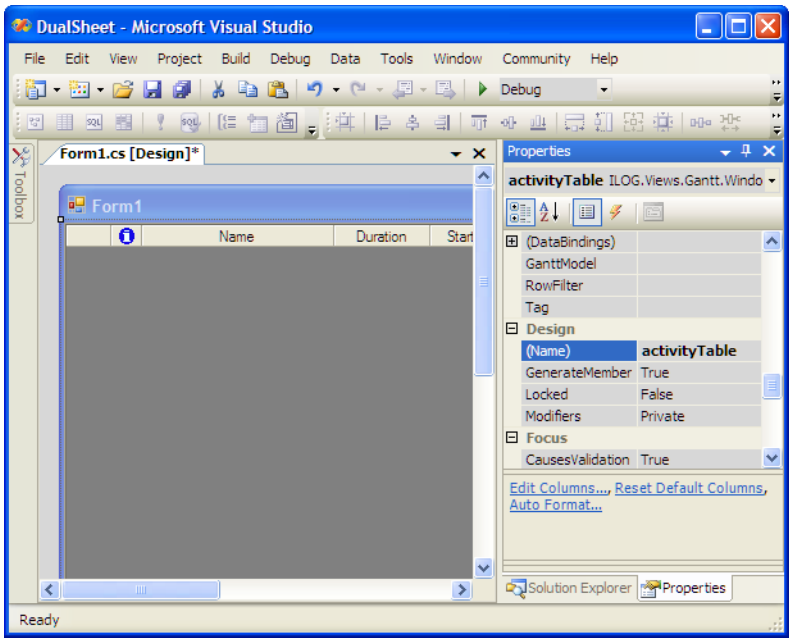

DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Form1.cs [Design]*
Form1
Name
Duration
Start
Toolbox
Properties
activityTable1 ILOG.Views.Gantt.Winc
Solution Explorer
Properties
Ready

Focus	
CausesValidation	True
Layout	
Anchor	Top, Left
ColumnHeaderHei	17
ColumnHeadersVis	True
Dock	**Fill**
HScrollBar	AsNeeded
Location	**0, 0**
Margin	3, 3, 3, 3
MaximumSize	0, 0

Edit Columns..., Reset Default Columns, Auto Format...

10. Rename the Activity Table.

Click the activity table in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **activityTable**.


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Name
Duration
Start
Properties
Solution Explorer
Properties
Ready

activityTable ILOG.Views.Gantt.Windo

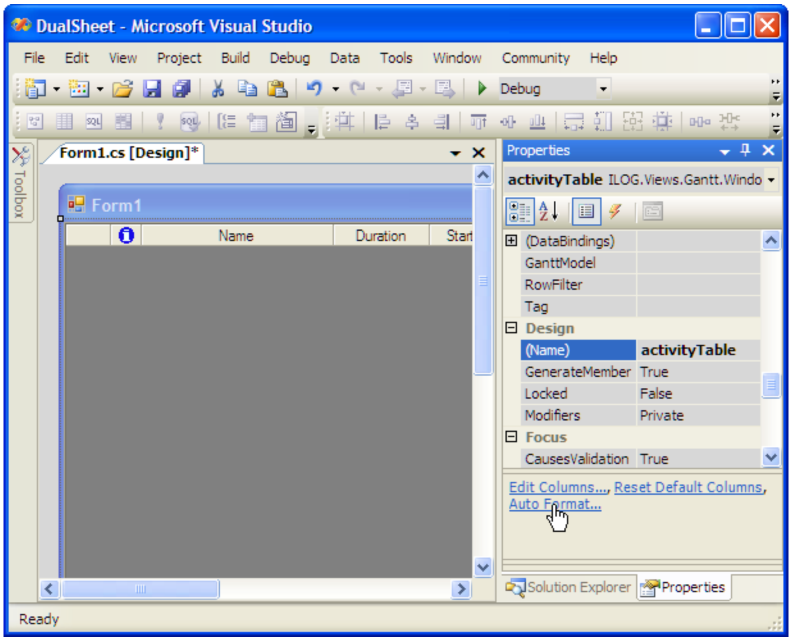
(DataBindings)	
GanttModel	
RowFilter	
Tag	
Design	
(Name)	activityTable
GenerateMember	True
Locked	False
Modifiers	Private
Focus	
CausesValidation	True

Edit Columns..., Reset Default Columns, Auto Format...

The appearance of the Activity Table can be completely customized to fit your needs.

11. Define a style for the Activity Table you have just created.

You can either right-click the activity table in Form1 and select AutoFormat in the contextual menu, or click AutoFormat at the bottom of the Properties explorer.

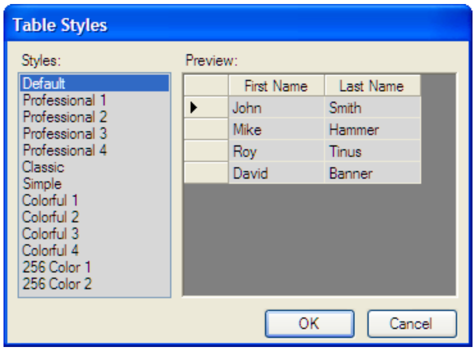

DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Form1.cs [Design]*
Form1
Name Duration Start
Properties
Solution Explorer
Properties
Ready

activityTable ILOG.Views.Gantt.Windo

(DataBindings)	
GanttModel	
RowFilter	
Tag	
Design	
(Name)	activityTable
GenerateMember	True
Locked	False
Modifiers	Private
Focus	
CausesValidation	True

Edit Columns..., Reset Default Columns, Auto Format...

The Table Styles window appears:



Styles:
Default
Professional 1
Professional 2
Professional 3
Professional 4
Classic
Simple
Colorful 1
Colorful 2
Colorful 3
Colorful 4
256 Color 1
256 Color 2

Preview:

First Name	Last Name
John	Smith
Mike	Hammer
Roy	Tinus
David	Banner

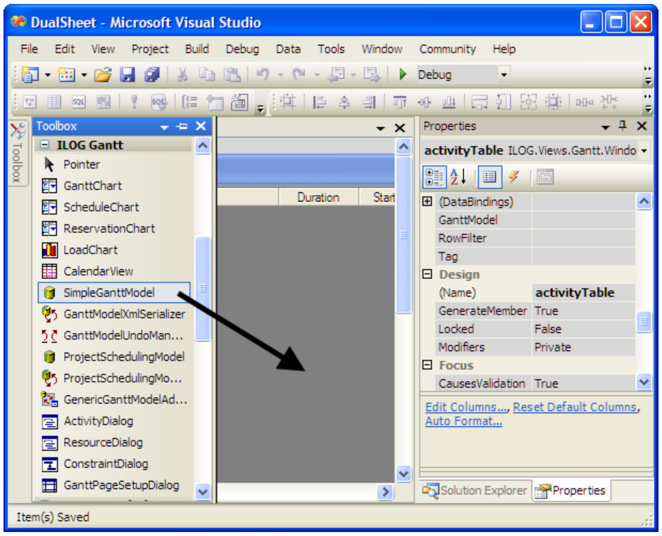
Select a style and click OK. The style that you selected is automatically applied to the Activity Table.

12. Save the file **Form1.cs**.

Associating a Gantt Model With the Activity Table

The Gantt data model contains the scheduling information you want to display in the activity table. In the following steps, you will learn how to bind a simple Gantt data model to the activity table.

1. From the IBM® ILOG® Gantt for .NET toolbox select the **SimpleGanttModel** object and drag it to Form1.

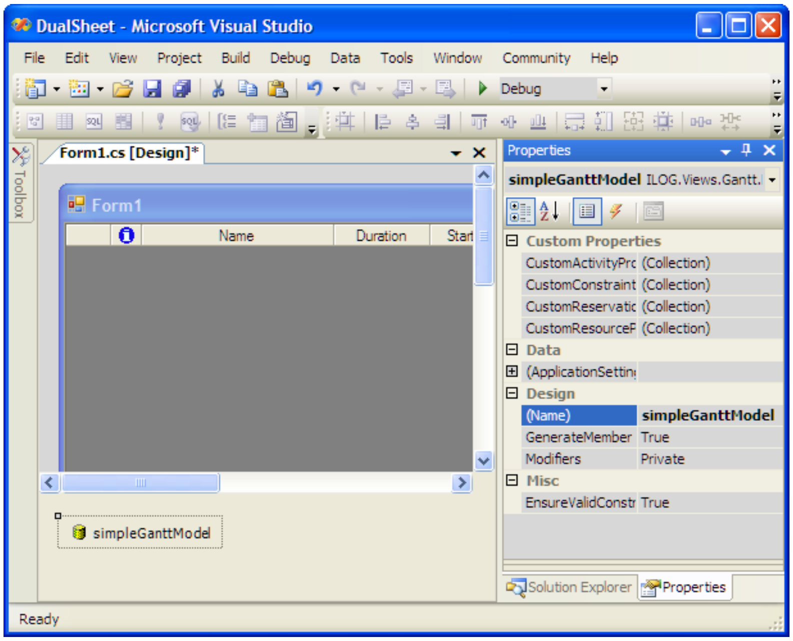


activityTable ILOG.Views.Gantt.Windo

(DataBindings)	
GanttModel	
RowFilter	
Tag	
Design	
(Name)	activityTable
GenerateMember	True
Locked	False
Modifiers	Private
Focus	
CausesValidation	True

2. Rename the simple Gantt model.

   Click the **SimpleGanttModel1** object in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **simpleGanttModel**.



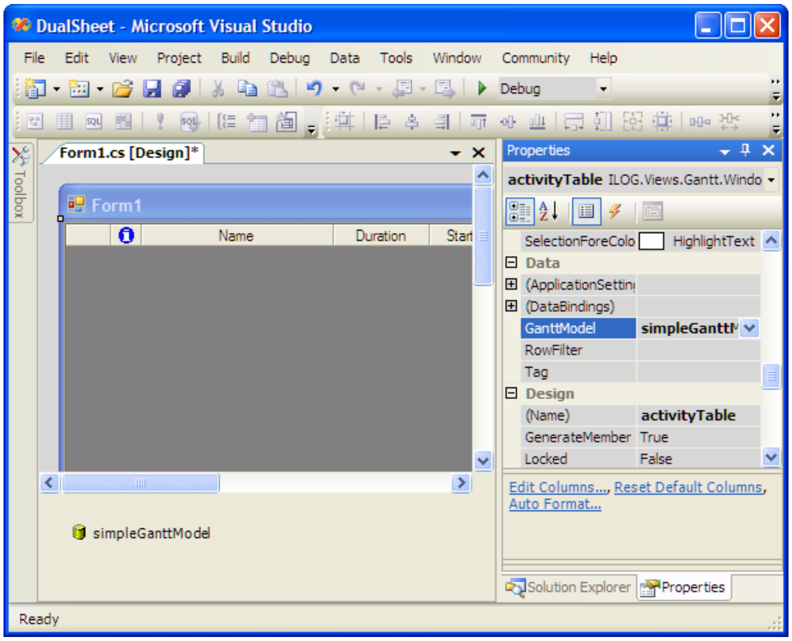
simpleGanttModel ILOG.Views.Gantt.

Custom Properties	
CustomActivityPrc	(Collection)
CustomConstraint	(Collection)
CustomReservatic	(Collection)
CustomResourceF	(Collection)
Data	
(ApplicationSettin	
Design	
(Name)	simpleGanttModel
GenerateMember	True
Modifiers	Private
Misc	
EnsureValidConstr	True

3. Associate the Gantt model with the Activity Table.

   Click the activity table in Form1 and from the Properties explorer select **Data > GanttModel**.

   Click the arrow of the drop-down list next to the GanttModel property and choose **simpleGanttModel**.



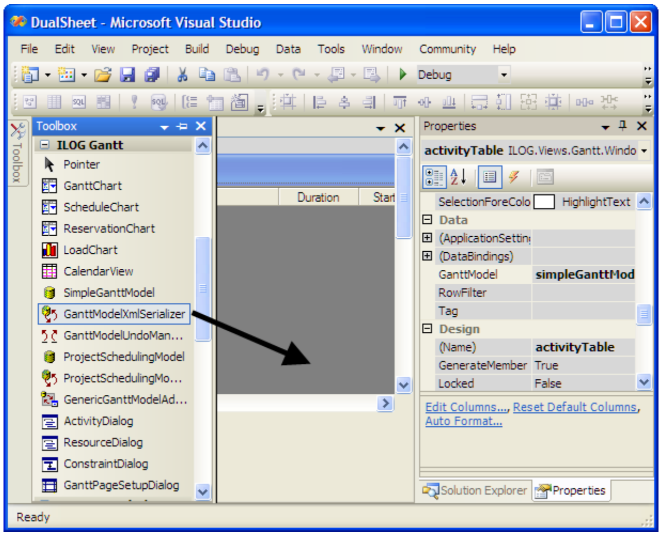
Now the simple Gantt model is associated with the activity table displayed in Form1.

Filling the Activity Table With XML Data

IBM® ILOG® Gantt for .NET allows you to serialize and deserialize scheduling data to and from Scheduling Data Exchange Language (SDXL) files. This is accomplished through the **GanttModelXmlSerializer** class.

To deserialize scheduling data from SDXL files and use this information to fill the Activity table, perform the following steps.

1. From the IBM ILOG Gantt for .NET toolbox select the **GanttModelXmlSerializer** object and drag it to Form1.



activityTable ILOG.Views.Gantt.Windo

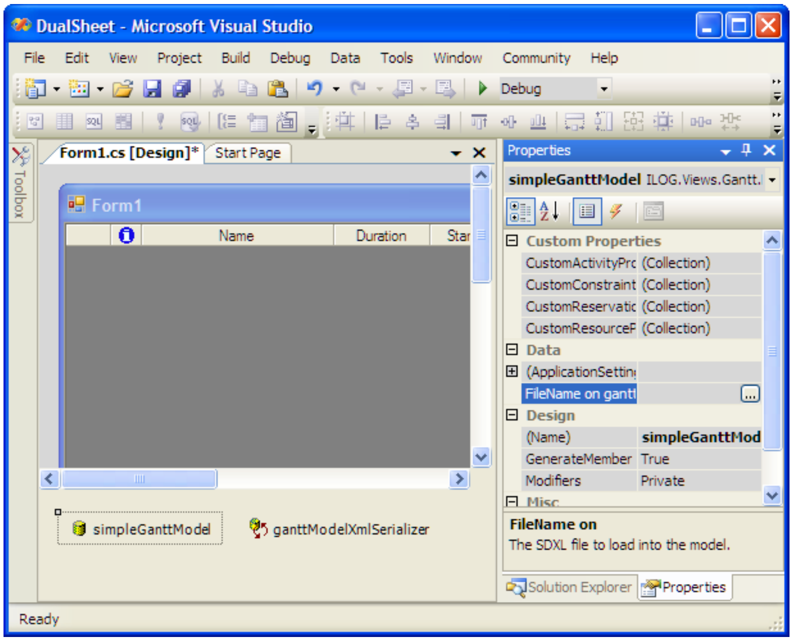
SelectionForeColo	HighlightText
Data	
(ApplicationSettin	
(DataBindings)	
GanttModel	simpleGanttMod
RowFilter	
Tag	
Design	
(Name)	activityTable
GenerateMember	True
Locked	False

2. Rename the Gantt model XML serializer.

Click the object **ganttModelXmlSerializer1** in Form1. From the Properties explorer choose **Design** > **Name** and enter the new name: **ganttModelXmlSerializer**.

3. Go back to **simpleGanttModel** and select it.

You will notice that a new property named **FileName on ganttModelXmlSerializer** has been created.



simpleGanttModel ILOG.Views.Gantt.

Custom Properties

CustomActivityPrc	(Collection)
CustomConstraint	(Collection)
CustomReservatic	(Collection)
CustomResourceF	(Collection)

Data

(ApplicationSettin	
FileName on gantt	

Design

(Name)	simpleGanttMod
GenerateMember	True
Modifiers	Private

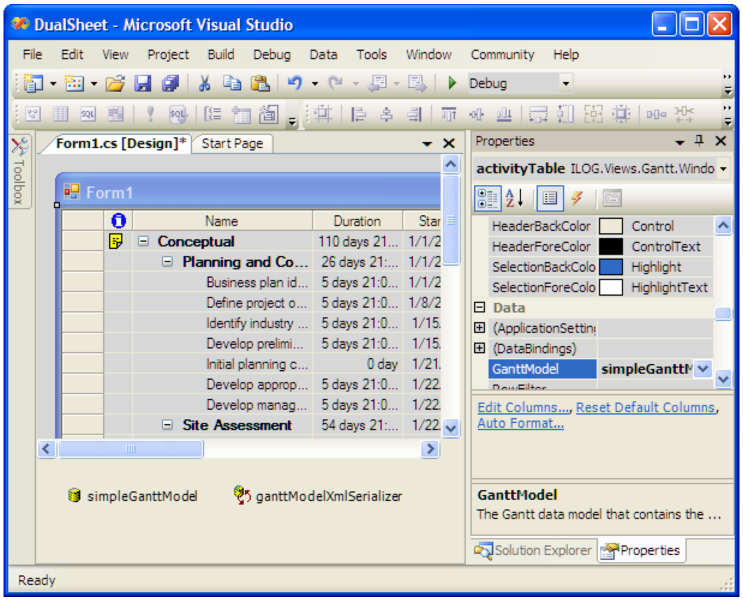
Misc

FileName on
The SDXL file to load into the model.

4. Select the property **FileName on ganttModelXmlSerializer** and click the three-dots button next to it. Choose an SDXL file to fill the activity table.

Note: *Examples of SDXL files can be found in <install-dir>\Data.*

Data contained in the SDXL file are displayed in the Activity Table.



		Name	Duration	Star
		⊟ **Conceptual**	110 days 21...	1/1/2
		⊟ **Planning and Co...**	26 days 21:...	1/1/2
		Business plan id...	5 days 21:0...	1/1/2
		Define project o...	5 days 21:0...	1/8/2
		Identify industry ...	5 days 21:0...	1/15.
		Develop prelimi...	5 days 21:0...	1/15.
		Initial planning c...	0 day	1/21.
		Develop approp...	5 days 21:0...	1/22.
		Develop manag...	5 days 21:0...	1/22.
		⊟ **Site Assessment**	54 days 21:...	1/22.

simpleGanttModel   ganttModelXmlSerializer

activityTable ILOG.Views.Gantt.Windo

HeaderBackColor	Control
HeaderForeColor	ControlText
SelectionBackColo	Highlight
SelectionForeColo	HighlightText
⊟ Data	
⊞ (ApplicationSettin	
⊞ (DataBindings)	
GanttModel	**simpleGanttM**

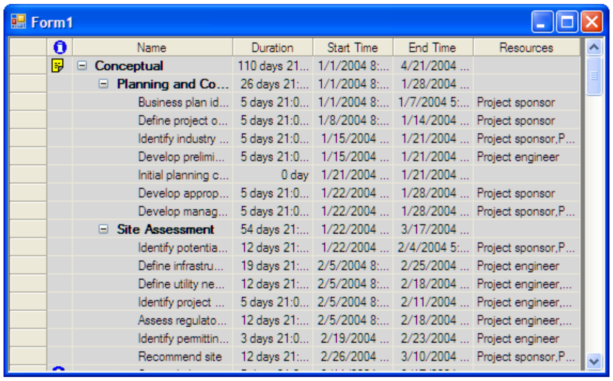
Edit Columns..., Reset Default Columns, Auto Format...

GanttModel
The Gantt data model that contains the ...

5. Save the file **Form1.cs**.

6. Run the application using the command **Debug** > **Start**.

   At this stage, the Activity Table contains only a table and appears as follows:



		Name	Duration	Start Time	End Time	Resources
		⊟ **Conceptual**	110 days 21...	1/1/2004 8:...	4/21/2004 ...	
		⊟ **Planning and Co...**	26 days 21:...	1/1/2004 8:...	1/28/2004 ...	
		Business plan id...	5 days 21:0...	1/1/2004 8:...	1/7/2004 5:...	Project sponsor
		Define project o...	5 days 21:0...	1/8/2004 8:...	1/14/2004 ...	Project sponsor
		Identify industry ...	5 days 21:0...	1/15/2004 ...	1/21/2004 ...	Project sponsor,P...
		Develop prelimi...	5 days 21:0...	1/15/2004 ...	1/21/2004 ...	Project engineer
		Initial planning c...	0 day	1/21/2004 ...	1/21/2004 ...	
		Develop approp...	5 days 21:0...	1/22/2004 ...	1/28/2004 ...	Project sponsor
		Develop manag...	5 days 21:0...	1/22/2004 ...	1/28/2004 ...	Project sponsor,P...
		⊟ **Site Assessment**	54 days 21:...	1/22/2004 ...	3/17/2004 ...	
		Identify potentia...	12 days 21:...	1/22/2004 ...	2/4/2004 5:...	Project sponsor,P...
		Define infrastru...	19 days 21:...	2/5/2004 8:...	2/25/2004 ...	Project engineer
		Define utility ne...	12 days 21:...	2/5/2004 8:...	2/18/2004 ...	Project engineer,...
		Identify project ...	5 days 21:0...	2/5/2004 8:...	2/11/2004 ...	Project engineer,...
		Assess regulato...	12 days 21:...	2/5/2004 8:...	2/18/2004 ...	Project engineer,...
		Identify permittin...	3 days 21:0...	2/19/2004 ...	2/23/2004 ...	Project engineer
		Recommend site	12 days 21:...	2/26/2004 ...	3/10/2004 ...	Project sponsor,P...

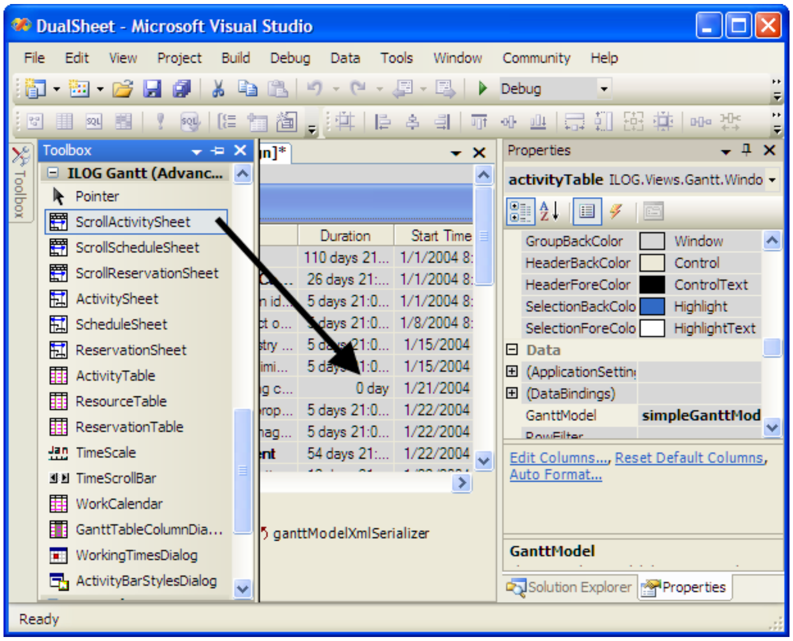
7. Close the application you have just executed.

Adding the Scroll Activity Sheets

Now you will learn how to associate and synchronize the scroll activity sheets with the table you have just created.

Adding the First Scroll Activity Sheet

1. From the IBM® ILOG® Gantt for .NET toolbox select the **ScrollActivitySheet** object and drag it to Form1.



activityTable ILOG.Views.Gantt.Windo

GroupBackColor	Window
HeaderBackColor	Control
HeaderForeColor	ControlText
SelectionBackColo	Highlight
SelectionForeColo	HighlightText
Data	
(ApplicationSettin	
(DataBindings)	
GanttModel	**simpleGanttMod**

GanttModel

2. Resize the scroll activity sheet.

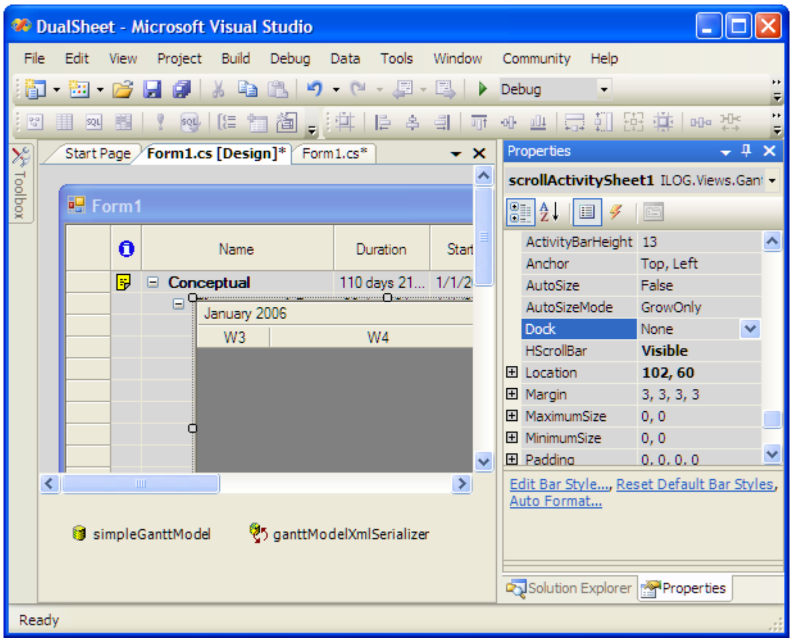
   Click the scroll activity sheet in Form1. From the Properties explorer choose **Layout** > **Size.** Expand the property and enter the following value:

   Width: 300

   Height: leave the default value.

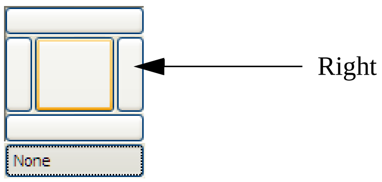
3. Change the Dock property to align the scroll activity sheet to the right of the Activity Table.

   Click the **scrollActivitySheet1** object in Form1 and from the Properties explorer choose **Layout** > **Dock**.


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Start Page Form1.cs [Design]* Form1.cs*
Form1
Name Duration Start
Conceptual 110 days 21... 1/1/2
January 2006
W3 W4
simpleGanttModel ganttModelXmlSerializer
Properties
scrollActivitySheet1 ILOG.Views.Gant
Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer Properties
Ready

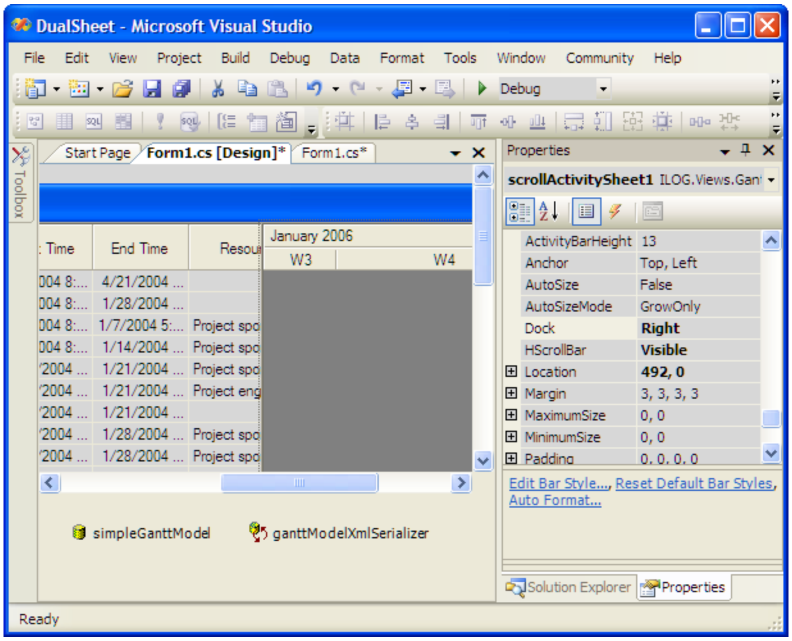
ActivityBarHeight	13
Anchor	Top, Left
AutoSize	False
AutoSizeMode	GrowOnly
Dock	None
HScrollBar	Visible
⊞ Location	102, 60
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
⊞ Padding	0, 0, 0, 0

Clicking the arrow next to the Dock property shows the following control:



4. Select the area corresponding to the Right option.

The scroll activity sheet is now aligned to the right of the Activity Table. If necessary, scroll horizontally to view it.


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Window Community Help
Debug
Toolbox
Start Page
Form1.cs [Design]*
Form1.cs*
January 2006
W3
W4
simpleGanttModel
ganttModelXmlSerializer
Properties
scrollActivitySheet1 ILOG.Views.Gan
Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

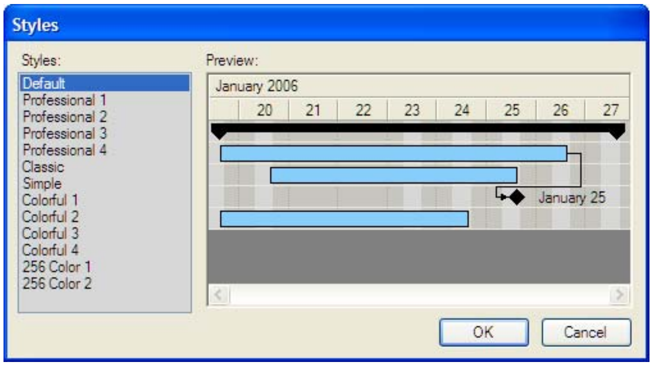
: Time	End Time	Resou
004 8:...	4/21/2004 ...	
004 8:...	1/28/2004 ...	
004 8:...	1/7/2004 5:...	Project spo
004 8:...	1/14/2004 ...	Project spo
'2004 ...	1/21/2004 ...	Project spo
'2004 ...	1/21/2004 ...	Project eng
'2004 ...	1/21/2004 ...	
'2004 ...	1/28/2004 ...	Project spo
'2004 ...	1/28/2004 ...	Project spo

ActivityBarHeight	13
Anchor	Top, Left
AutoSize	False
AutoSizeMode	GrowOnly
Dock	**Right**
HScrollBar	**Visible**
⊞ Location	**492, 0**
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
⊞ Padding	0, 0, 0, 0

5. Define a style for the scroll activity sheet you have just created.

You can either right-click the scroll activity sheet in Form1 and select AutoFormat in the contextual menu, or click AutoFormat at the bottom of the Properties explorer.

The Styles window appears:

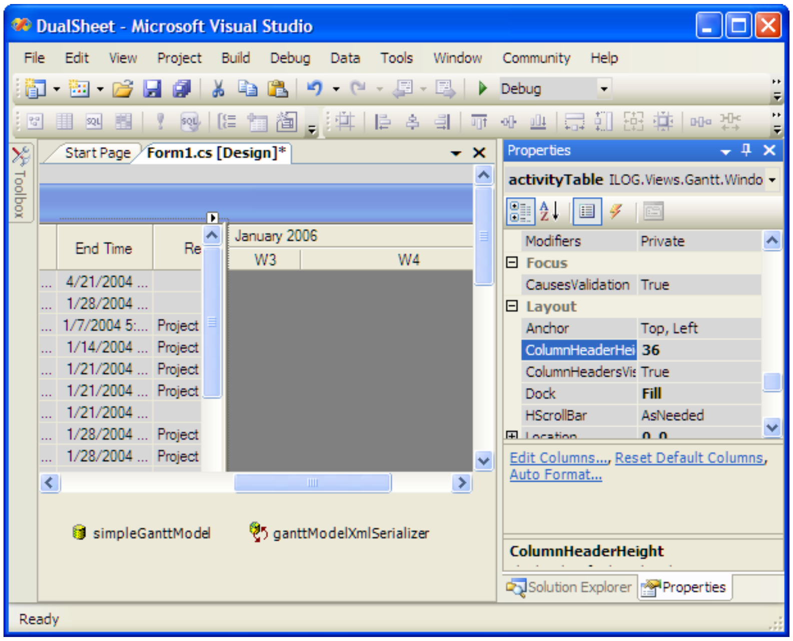


Select a style and click OK. The style that you selected is automatically applied to the scroll activity sheet.

6. Go back to the activity table. Right-click it and choose Bring to Front.

7. Synchronize the table header with the time scale.

With the activity table selected, from the Properties explorer choose **Layout > ColumnHeaderHeight** and enter 36.

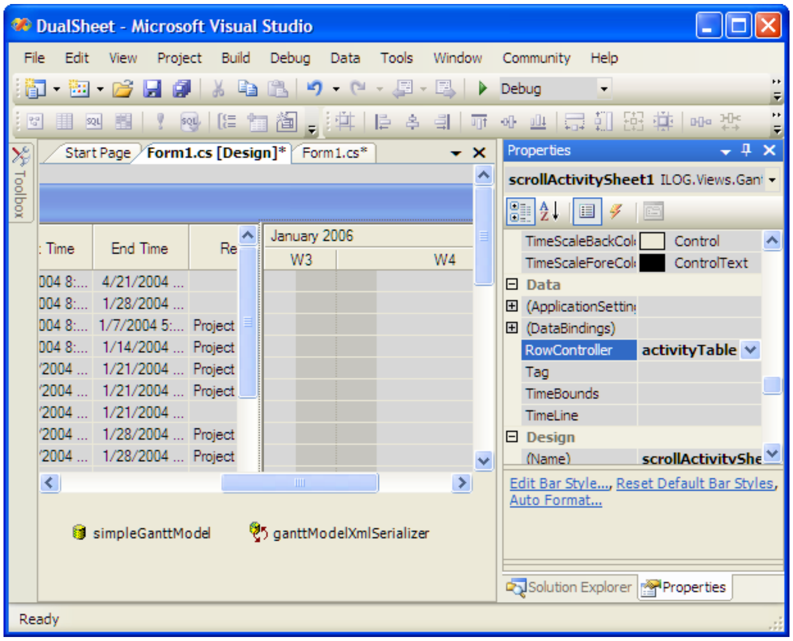

DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Start Page
Form1.cs [Design]*
January 2006
W3
W4
simpleGanttModel
ganttModelXmlSerializer
Properties
activityTable ILOG.Views.Gantt.Windo
Edit Columns..., Reset Default Columns, Auto Format...
ColumnHeaderHeight
Solution Explorer
Properties
Ready

	End Time	Re
...	4/21/2004 ...	
...	1/28/2004 ...	
...	1/7/2004 5:...	Project
...	1/14/2004 ...	Project
...	1/21/2004 ...	Project
...	1/21/2004 ...	Project
...	1/21/2004 ...	
...	1/28/2004 ...	Project
...	1/28/2004 ...	Project

Modifiers	Private
⊟ Focus	
CausesValidation	True
⊟ Layout	
Anchor	Top, Left
ColumnHeaderHei	36
ColumnHeadersVis	True
Dock	Fill
HScrollBar	AsNeeded
⊞ Location	0, 0

8. Synchronize vertically the activity table with the scroll activity sheet.

With the scroll activity sheet selected, from the Properties explorer choose **Data > RowController**. Click the arrow of the drop-down list next to the RowController property and choose **activityTable**.


DualSheet - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
Toolbox
Start Page
Form1.cs [Design]*
Form1.cs*
Properties
simpleGanttModel
ganttModelXmlSerializer
Edit Bar Style..., Reset Default Bar Styles, Auto Format...
Solution Explorer
Properties
Ready

: Time	End Time	Re
004 8:...	4/21/2004 ...	
004 8:...	1/28/2004 ...	
004 8:...	1/7/2004 5:...	Project
004 8:...	1/14/2004 ...	Project
'2004 ...	1/21/2004 ...	Project
'2004 ...	1/21/2004 ...	Project
'2004 ...	1/21/2004 ...	
'2004 ...	1/28/2004 ...	Project
'2004 ...	1/28/2004 ...	Project

January 2006 — W3, W4

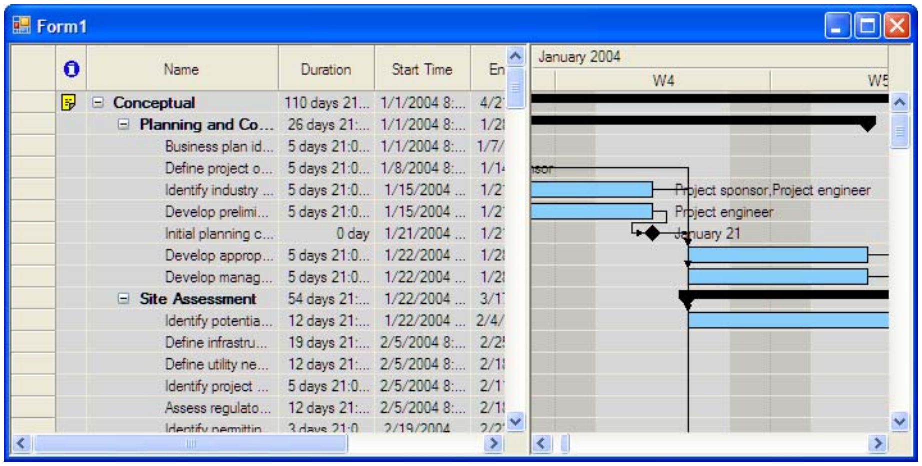
scrollActivitySheet1 ILOG.Views.Gant

TimeScaleBackCol	Control
TimeScaleForeCol	ControlText
Data	
(ApplicationSettin	
(DataBindings)	
RowController	**activityTable**
Tag	
TimeBounds	
TimeLine	
Design	
(Name)	**scrollActivityShe**

9. Save the file Form1.cs.

10. Run the application using the command **Debug** > **Start**.

At this stage, the Gantt chart application contains a table and a sheet, and appears as follows:

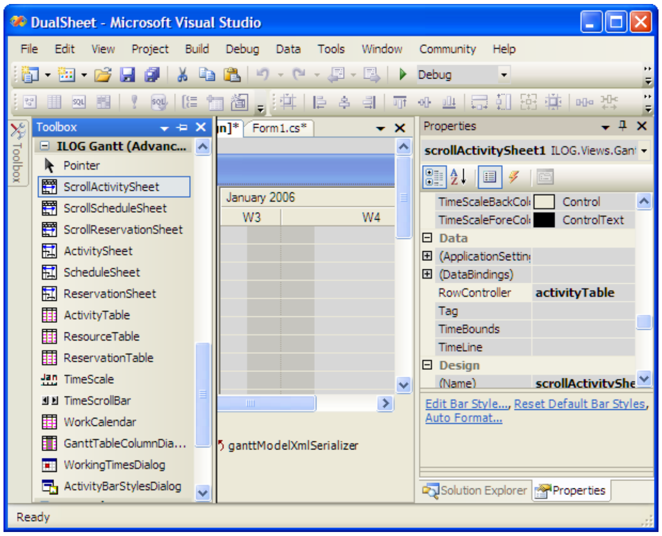


Name	Duration	Start Time	En
Conceptual	110 days 21...	1/1/2004 8:...	4/2
Planning and Co...	26 days 21:...	1/1/2004 8:...	1/2
Business plan id...	5 days 21:0...	1/1/2004 8:...	1/7/
Define project o...	5 days 21:0...	1/8/2004 8:...	1/1
Identify industry ...	5 days 21:0...	1/15/2004 ...	1/2
Develop prelimi...	5 days 21:0...	1/15/2004 ...	1/2
Initial planning c...	0 day	1/21/2004 ...	1/2
Develop approp...	5 days 21:0...	1/22/2004 ...	1/2
Develop manag...	5 days 21:0...	1/22/2004 ...	1/2
Site Assessment	54 days 21:...	1/22/2004 ...	3/1
Identify potentia...	12 days 21:...	1/22/2004 ...	2/4/
Define infrastru...	19 days 21:...	2/5/2004 8:...	2/2
Define utility ne...	12 days 21:...	2/5/2004 8:...	2/1
Identify project ...	5 days 21:0...	2/5/2004 8:...	2/1
Assess regulato...	12 days 21:...	2/5/2004 8:...	2/1
Identify permittin...	3 days 21:0...	2/19/2004 ...	2/2

11. Close the application you have just executed.

Adding the Second Scroll Activity Sheet

1. From the IBM ILOG Gantt for .NET toolbox select the **ScrollActivitySheet** object and drag it to Form1.



scrollActivitySheet1 ILOG.Views.Gan

TimeScaleBackCol	Control
TimeScaleForeCol	ControlText
Data	
(ApplicationSettin	
(DataBindings)	
RowController	activityTable
Tag	
TimeBounds	
TimeLine	
Design	
(Name)	scrollActivitySheet

2. Resize the second scroll activity sheet.

Click the scroll activity sheet in Form1. From the Properties explorer choose **Layout** > **Size**, expand the property and enter the following value:

Width: 300

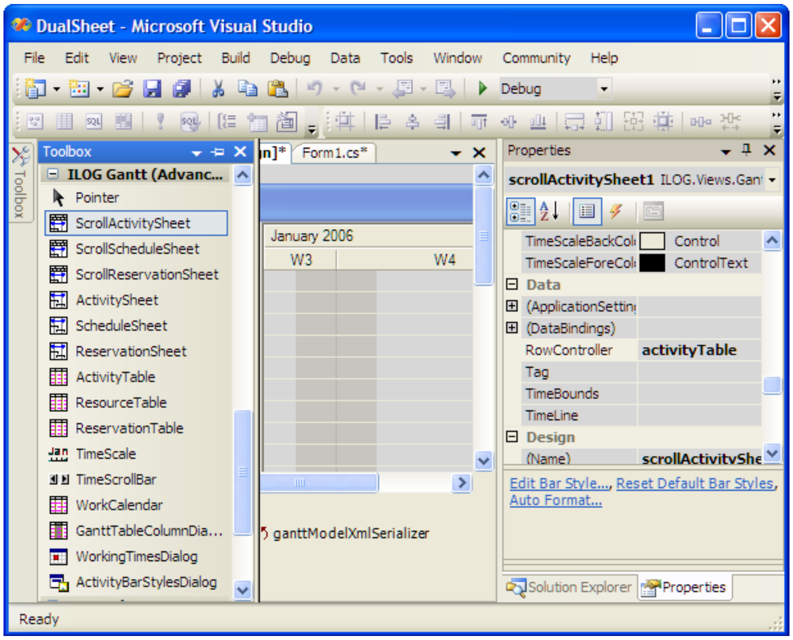
Height: leave the default value.

3. Change the Dock property to align the scroll activity sheet to the right of the activity table.

Click the second scroll activity sheet in Form1 and from the Properties explorer choose **Layout** > **Dock**.

4. Select the area corresponding to the Right option.

The second scroll activity sheet is now aligned to the right of the activity table.

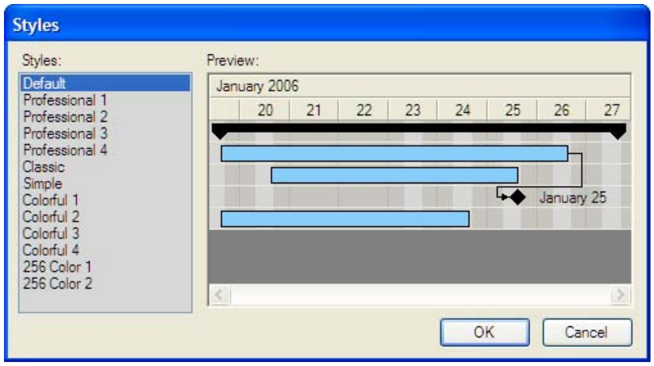


TimeScaleBackCol	Control
TimeScaleForeCol	ControlText
Data	
(ApplicationSettin	
(DataBindings)	
RowController	activityTable
Tag	
TimeBounds	
TimeLine	
Design	
(Name)	scrollActivityShe

5. Define a style for the scroll activity sheet you have just created.

   You can either right-click the scroll activity sheet in Form1 and select AutoFormat in the contextual menu, or click AutoFormat at the bottom of the Properties explorer.

   The Styles window appears:



   Select a style and click OK. The style that you selected is automatically applied to the scroll activity sheet.

6. Go back to the activity table. Right-click it and choose Bring to Front.

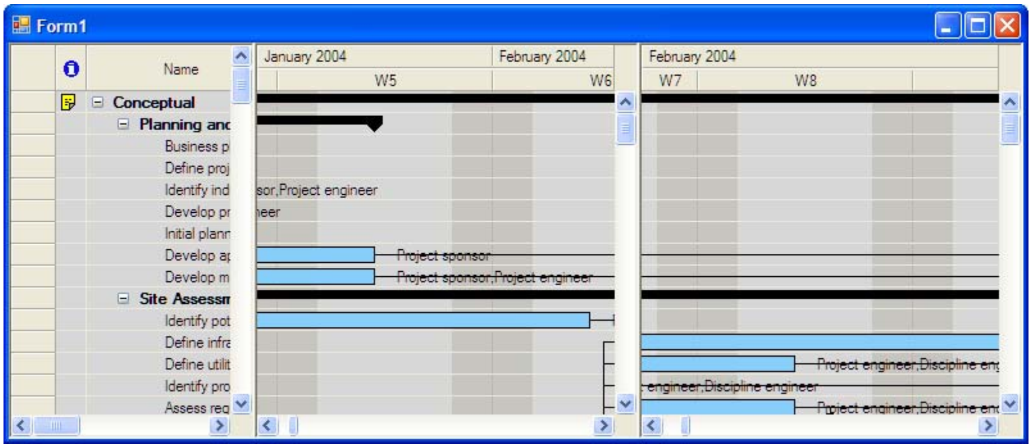
7. Synchronize vertically the Activity Table with the second scroll activity sheet.

   With the second scroll activity sheet selected, from the Properties explorer choose **Data > RowController**. Click the arrow of the drop-down list next to the RowController property and choose **activityTable**.

8. Save the file **Form1.cs**.

9. Run the application using the command **Debug** > **Start**.

   Now that you have added the second scroll activity sheet, the application appears as follows:



10. Close the application you have just executed.

Making Reservation Visible at Launch Time

When the application is launched, the scroll activity sheets appear empty because their default visible time interval may not contain data, depending on the SDXL file you want to load. The scheduling information is there but you have to scroll to view it. The following steps illustrate how to make the reservations immediately visible in the sheets when the application is launched. The first scroll activity sheet will display the first half of the scheduling information, and the second scroll activity sheet will display the second half.

1. In Design mode, double-click the header of Form1.

   The code editor opens with the cursor placed automatically at the appropriate location.

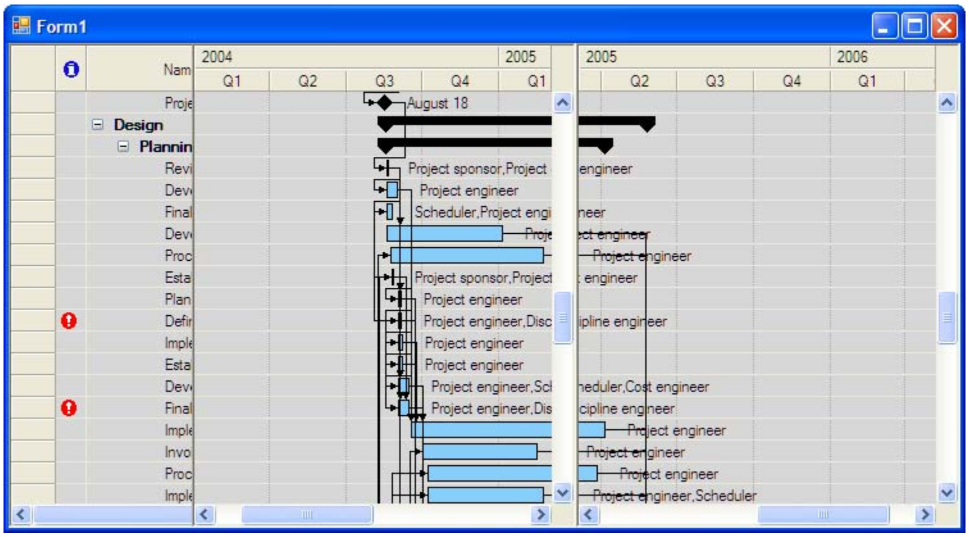
2. Insert the following code:

```
ILOG.Views.Gantt.DateTimeInterval interval =
scrollActivitySheet1.TimeBounds.TimeInterval;
scrollActivitySheet2.VisibleTimeInterval = new
ILOG.Views.Gantt.DateTimeInterval(interval.Start,
TimeSpan.FromTicks((long)(interval.Duration.Ticks / 2)));
scrollActivitySheet1.VisibleTimeInterval = new
ILOG.Views.Gantt.DateTimeInterval(scrollActivitySheet2.LastVisibleTime,
TimeSpan.FromTicks((long)(interval.Duration.Ticks / 2)));
```

3. Save the file **Form1.cs**.

4. Run the application using the command **Debug** > **Start**.

Now the reservations are immediately visible in the table when the application is launched, as illustrated below:



Index

A

activity table **55**

G

Gantt chart **1**, **15**
Gantt Sheet **1**, **15**

L

load chart **42**

R

resource **49**

S

schedule chart **27**
SDXL **11**, **20**
splitter **48**

T

time line **46**
tutorials
  adding load chart **42**
  adding scroll activity sheets **70**
  associating Gantt data model with Gantt chart **10**, **20**
  associating Gantt data model with schedule chart **36**
  associating Gantt model with activity table **64**
  creating a schedule chart **27**
  creating activity table **55**
  creating the Gantt chart **1**, **15**
  making reservation visible at launch time **77**
  making reservations visible at launch time **51**
  synchronizing charts splitter **48**
  synchronizing resource selection **49**
  synchronizing the charts **46**
  synchronizing visible time range **46**