IBM ILOG Gantt for .NET V4.0

Getting Started

June 2009

Table of Contents

Getting Started

For an accelerated introduction to the IBM® ILOG® Gantt for .NET Framework, follow these steps in sequence:

1. Read the *Overview of IBM ILOG Gantt for .NET* topic.
2. Scan the Tutorials, beginning with the *Creating a Windows Forms Gantt Application* tutorial.
3. Identify the samples that are relevant to your development needs by using the Sample Browser (Windows® Start Menu in All Programs>IBM ILOG>IBM ILOG Gantt for .NET>Sample Browser) which allows you to quickly scan what the product can offer you. The samples are installed in <*install-dir*>\Samples. Use the samples as a foundation for developing your own Gantt applications.
4. As you begin to develop your applications, continue to use the IBM ILOG Gantt for .NET documentation as a primary source of information. The documentation is compiled into an easy-to-use Help file that is displayed in the Microsoft® Document Explorer viewer. The viewer includes features such as an index keyword search, a full-text search, that enable you to customize how the information is displayed.

In This Section

What's New in IBM ILOG Gantt for .NET 4.0

Describes the new features and new documentation for version 4.0.

Overview of IBM ILOG Gantt for .NET

Introduces the IBM ILOG Gantt for .NET architecture and its components.

Document Conventions

Shows the typographic conventions for programming elements that are referenced in the documentation.

System Requirements

Lists the minimum and recommended system requirements for client and server applications.

License Management

Provides information about license requirements.

Support

Provides information about product support on the IBM ILOG Gantt for .NET SDK.

What's New in IBM ILOG Gantt for .NET 4.0

IBM® ILOG® Gantt for .NET version 4.0 extends version 3.5 with new features, improvements to existing features and enhancements to the documentation.

This section provides information about the key additions and modifications.

In This Section

New IBM ILOG Gantt for .NET Support for Silverlight

Describes how Silverlight is supported within IBM ILOG Gantt for .NET 4.0.

Compatibility Issues

Describes the changes that could affect compatibility with the previous version.

New IBM ILOG Gantt for .NET Support for Silverlight

IBM® ILOG® Gantt for .NET 4.0 offers the ability to integrate Gantt chart displays inside your Silverlight application.The Silverlight support brings a dedicated set of Silverlight elements that ease the creation of Gantt chart representations. With these classes, you may create your Gantt chart directly from any data source, by specifying the activities, resources, constraints and reservations that compose your Gantt model. Thanks to the power of Silverlight, it is possible to create very appealing Gantt chart representations through the

styling and templating features of Silverlight. The ProjectSchedulingModel is also available to Silverlight applications.

For more details on the Silverlight support, see *Programming with IBM ILOG Gantt for .NET Silverlight Controls*.

Compatibility Issues

Visual Studio Support

IBM® ILOG® Gantt for .NET no longer supports Visual Studio 2005 anymore. It's still possible to use Visual Studio 2005 to build .NET 2.0 projects, but the integration inside Visual Studio is not provided, and the sample projects for this platform are no longer shipped.

Dependency Change for Web Controls

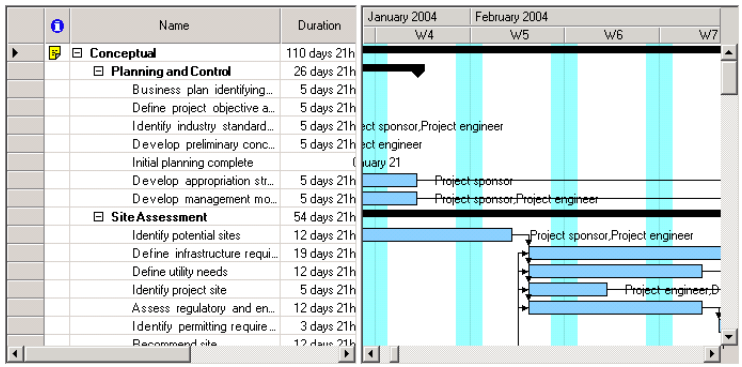
To build AJAX enabled applications, IBM ILOG Gantt for .NET 4.0 requires by default the .NET Framework 3.5 SDK. In case you want to target a .NET Framework 2.0 Web application and ASP.NET AJAX 1.0, IBM ILOG Gantt for .NET 4.0 provides the **ILOG.Gantt.Web.Ajax10.dll** assembly which has a dependency with ASP.NET AJAX 1.0. Note that in this case, the Visual Studio toolbox integration must be performed manually.

Overview of IBM ILOG Gantt for .NET

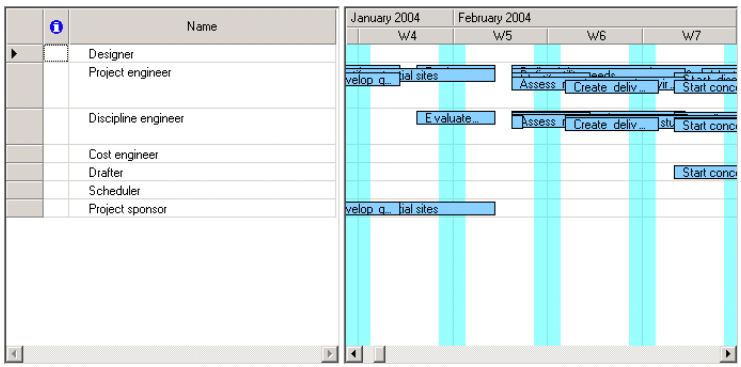
IBM® ILOG® Gantt for .NET consists of a library of classes meant to display and to edit a scheduling data model as a Gantt chart deployed in a Windows® Forms application, in an ASP.NET application, or in Internet Explorer.

It supports the two essential ways of displaying schedules: activity-oriented and resource oriented, so that it can be used in a large variety of applications and industries: transportation scheduling and planning, logistics, supply chain management, production scheduling and planning, workforce planning, resource booking, project management, and many more.

The following illustration shows an activity-oriented chart for a project management application.



The following illustration shows a resource-oriented chart for a production scheduling application.



IBM ILOG Gantt for .NET enables rapid application development with pre-built Windows® Forms and Web Forms controls that can be easily assembled and connected in Visual Studio 2008. It also offers deep customization capabilities, with a comprehensive set of documented classes designed to be extended in order to meet the most demanding applications requirements.

An innovative styling mechanism enables fast tailoring of the Gantt displays to present the most relevant scheduling information with the most intuitive means and insure great user experience.

Beyond displaying and editing scheduling information, IBM ILOG Gantt for .NET enables also printing with various layout parameters and a print preview.

The Gantt architecture has been designed with a clear separation between the data model and its visualization, following a Model-View-Controller (MVC) design pattern.

The next sections introduce the data model and the controls.

In This Section

The Data Model

Describes the scheduling data contained in the data model.

The Controls

Presents the graphical representation part of the Gantt class library.

Model-View Separation

Explains the MVC model, or how the data model can be shared between several views.

The Services

Presents the printing and styling services.

Related Sections

Programming with IBM ILOG Gantt for .NET Windows Forms and Web Forms Controls

Describes the essential programming information you need to build applications with IBM ILOG Gantt for .NET.

Building Applications with IBM ILOG Gantt for .NET

Provides instructive overviews and detailed step-by-step procedures for creating particular categories of applications, such as ASP.NET applications and Windows® Forms applications.

Tutorials

Provides step-by-step introductions to the fundamental areas of programming for the IBM ILOG Gantt for .NET.

The Data Model

The data model is the part of the library that contains the scheduling information you want to display.

As no scheduling application share the same data model, IBM® ILOG® Gantt for .NET provides an abstract data model that can be implemented to satisfy any possible case (XML, database, objects), and all memory or performance constraints (data duplication or not, load-on-demand or not). In the product is delivered a pre-built implementation that may satisfy many needs and that will accelerate development.

Scheduling data consists of four abstract entities:

- *Activities*
- *Constraints*
- *Resources*
- *Reservations*

IBM ILOG Gantt for .NET comes with default implementations of these entities. However, if these are not suited to your particular application, you can create your own user-defined entities/implementations.

Activities

An *activity* is a task that must be completed. Activities are hierarchical in nature. This means that a main activity, called *parent activity,* can be broken down into subactivities, called child activities.

In addition to its name and identifier, an activity is defined by its *start time* and *end time*, which determine an interval called the duration of the activity. If the start and end times are identical, the duration is equal to 0. A zero-duration activity is commonly called a *milestone*. Typically, milestones are not rendered by the same graphic object as activities with a non-zero duration.

Constraints

A *constraint* is a type of condition set between two activities. Constraints can have one of the following types: *start to start*, *start to end*, *end to start*, or *end to end*. The source activity (that is, the activity whose start or end controls the start or end of another activity) is called the *From activity.* Conversely, the target activity-that is, the activity whose start or end depends on the start or end of another activity-is called the *To activity.*

Constraints are represented by arrowed polyline links.

Resources

A *resource* enables an activity to be completed. Resources can be persons, premises, equipment, and so forth. Like activities, resources are also hierarchical in nature. For example, if resources are people, the *parent resource* is a department while the child resources are the individual employees. Likewise, you may want to group resources by physical location or by type of machinery.

Reservations

When a resource is assigned to an activity, this assignment is called a *reservation*. In the terminology of IBM ILOG Gantt for .NET, a reservation represents the assignment of one resource to one activity. An activity can have multiple resources reserved and similarly, a resource can be reserved for more than one activity.

See Also *Creating and Using Gantt Data Models*

The Controls

Using the full power of the IBM® ILOG® Gantt for .NET, you can view a data model containing scheduling information through a wide variety of graphical representations.

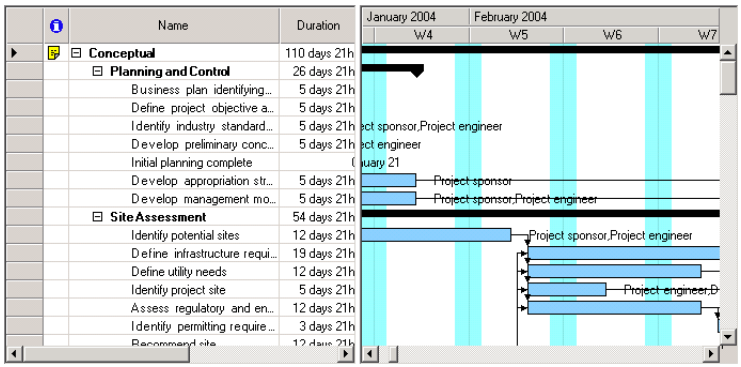
IBM ILOG Gantt for .NET offers a set of Windows® Forms and Web Forms controls that can be combined to meet your presentation requirements, and make sure that the right information is conveyed to the user efficiently and intuitively.

The Gantt Chart and the Schedule Chart controls encapsulate the most commonly used scheduling displays. You are also able to create your own presentation through combining

lower level controls such as hierarchical tables, time scales, load charts, calendar, Gantt sheets and schedule sheets.

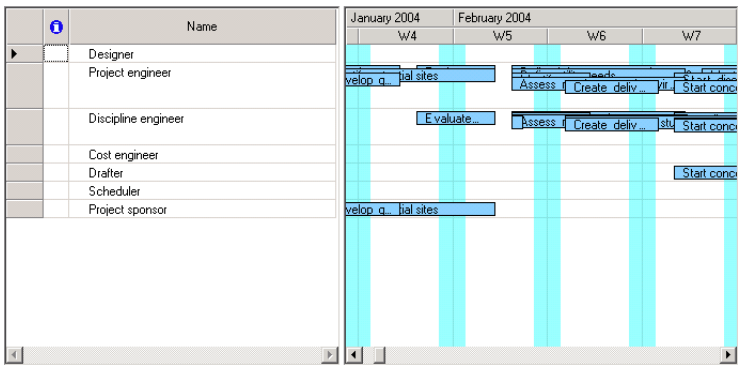
Gantt Chart for Activity-Oriented Presentations

The Gantt chart presentation shows one activity in each row. The hierarchy table on the left displays activity information from the data model. The Gantt sheet on the right shows how the activities are positioned on the time scale.



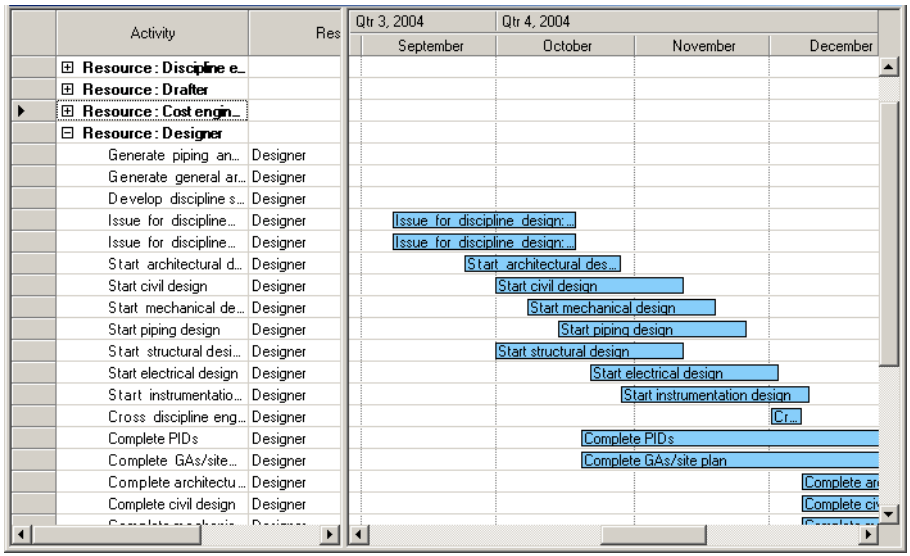
Schedule Chart for Resource-Oriented Presentations

The schedule chart presentation shows one resource in each row. The resource table on the left displays resource information from the data model. The schedule sheet on the right shows the resource reservations. Each row in the schedule sheet contains 0, 1, or more bars to represent the activities for which the matching resource has been reserved.



Reservation Chart for Resource-Oriented Presentations

The reservation chart presentation shows one reservation in each row. The reservation table on the left displays reservation information from the data model. The reservation sheet on the right shows the reservations. Each row in the reservation sheet contains 0, 1, or more bars to represent the activities used by reservations.

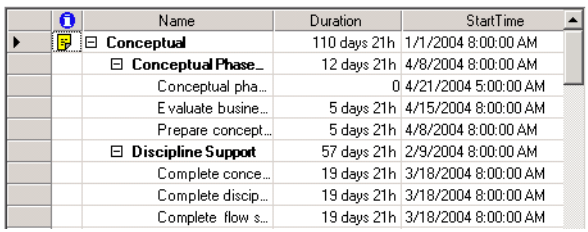


Individual Controls

The above Gantt and schedule chart controls are built upon lower-level controls that you can combine to create your own presentation. Through Visual Studio 2008, you can drag and drop the controls and connect them altogether. The synchronization between the controls is achieved through the data model and the time line object.

The following illustrations show the lower-level controls:

- Activity Table



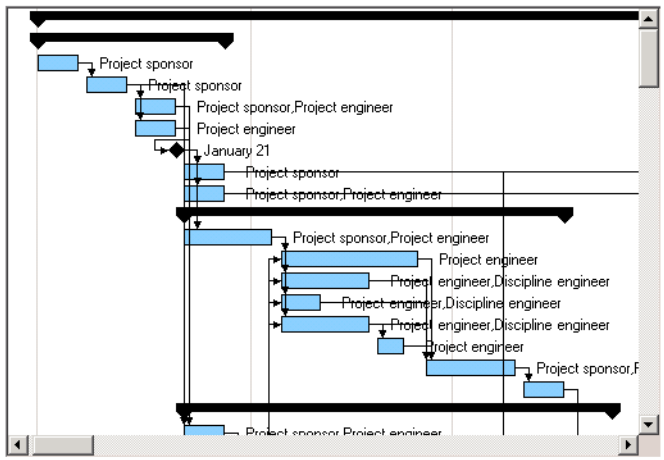
- Resource Table

	Name	MaxUnits	Note
▶	Designer	100 %	
	Project engineer	100 %	
	Discipline engineer	100 %	
	⊟ **Cost engineer**	100 %	
	David	100 %	
	Engineer 1	100 %	
	Engineer 2	100 %	

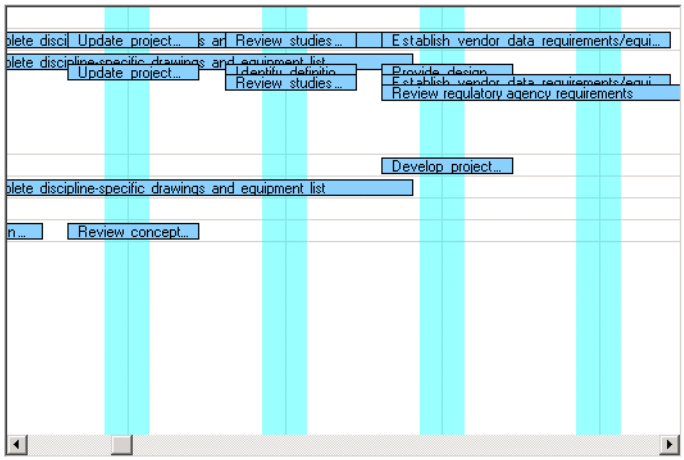
- Reservation Table

	Activity	Resource	Units
▶	Business plan identifyin...	Project sponsor	1
	Define project objective...	Project sponsor	1
	Identify industry standar...	Project sponsor	1
	Identify industry standar...	Project engineer	1
	Develop preliminary con...	Project engineer	1
	Develop appropriation s...	Project sponsor	1
	Develop management...	Project sponsor	1
	Develop management...	Project engineer	1
	Identify potential sites	Project sponsor	1
	Identify potential sites	Project engineer	1
	Define infrastructure req...	Project engineer	1

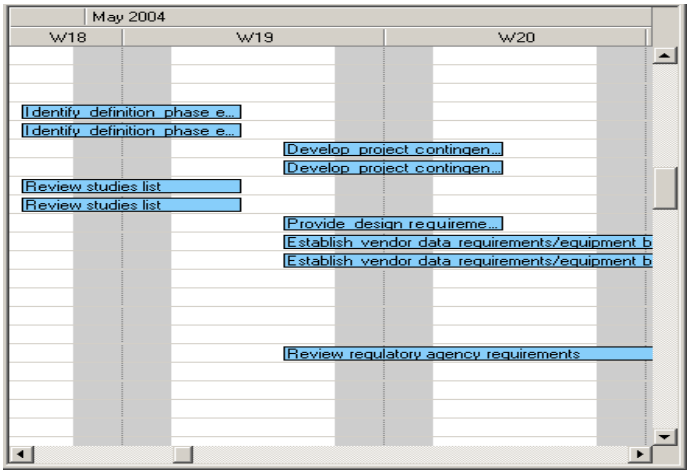
- Activity Sheet



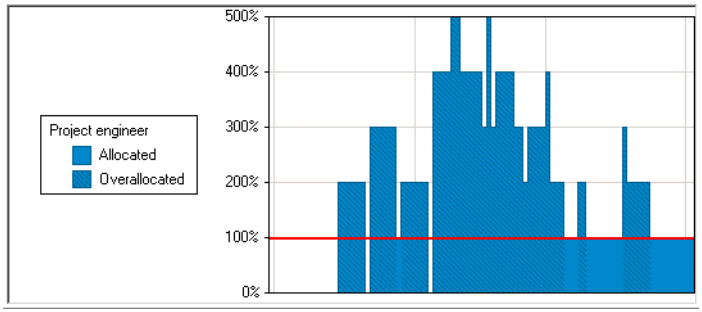
- Resource Sheet



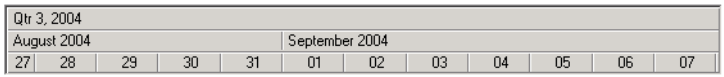
- Reservation Sheet



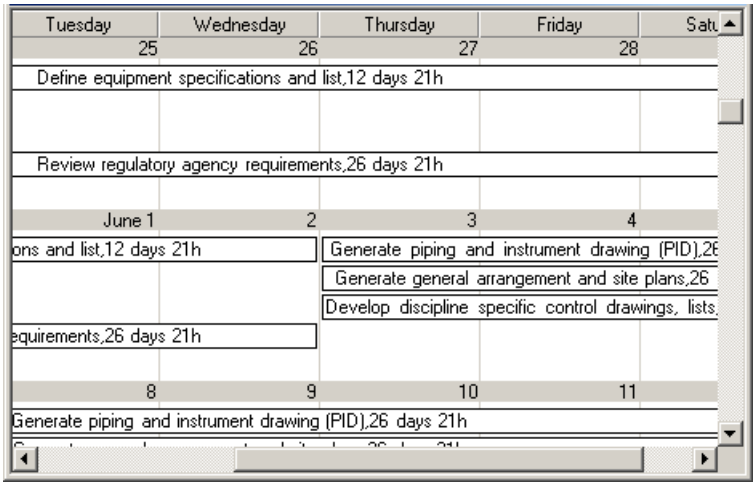
- Load Chart



- Time Scale



- Calendar View



See Also *Displaying Scheduling Data Using Gantt Charts* | *Creating Custom Gantt Representations*

Model-View Separation

IBM® ILOG® Gantt for .NET is based on a Model View Controller (MVC) architecture. The library contains an abstract data model and several controls for displaying the data model with different representations. Each control listens for data model events to be updated when the model is modified.

The traditional design of user-interface objects is referred to as MVC architecture because it divides each component into three parts:

- Model
- View
- Controller

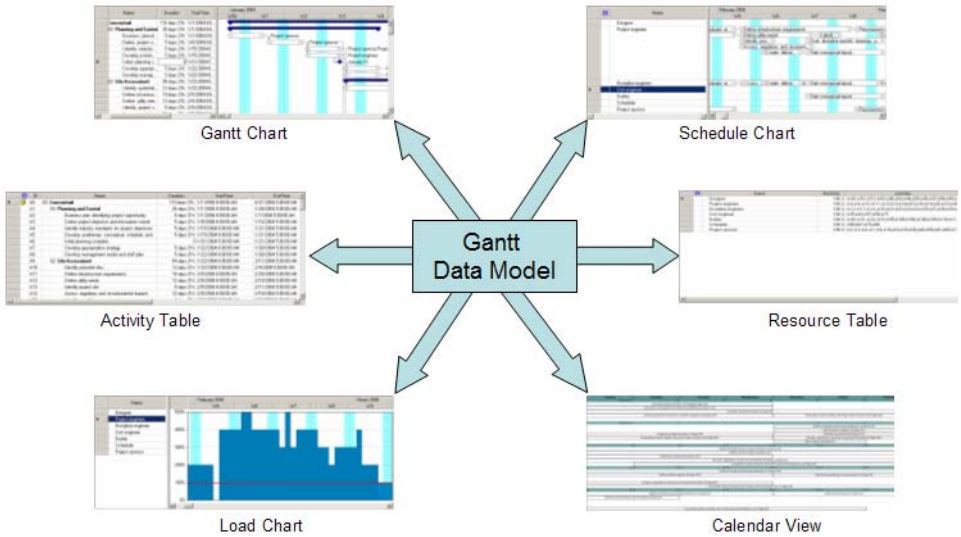
In this classic design, the model manages the data or values represented by the component, the view manages the way the component is displayed, and the controller handles user interaction with the component.

IBM ILOG Gantt for .NET is based on a variant of MVC called separable model architecture. This design provides all the benefits of complete model-view separation while being easier to use, because it bundles the view and controller parts together.

IBM ILOG Gantt for .NET defines an abstract scheduling data interface. This interface or one of its concrete implementations is referred to as the Gantt data model.

IBM ILOG Gantt for .NET also defines several controls that can be bound to a Gantt data model.

The following illustration shows the separable model architecture of IBM ILOG Gantt for .NET.



The Services

IBM® ILOG® Gantt for .NET brings ready-to-use services for more customizability and faster development.

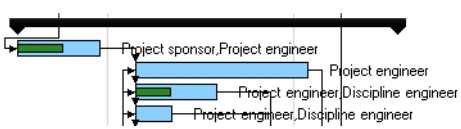
Styling

Schedule displays need to be adapted to the final user needs by choosing the information that is relevant to show and by selecting the best visual way to convey the information up to the user.

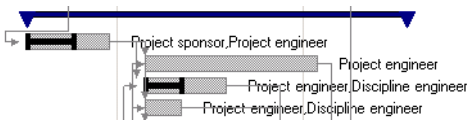
A styling mechanism is included in IBM ILOG Gantt for .NET to rapidly adapt the presentations to the user needs. Activities are mapped to bar styles according to their properties. Expressions let you determine the conditions for applying a style, and several bar styles can be combined to display a single activity.

The following illustrations show the different styles you can apply to the same data model.

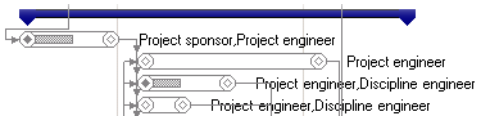
- Style 1



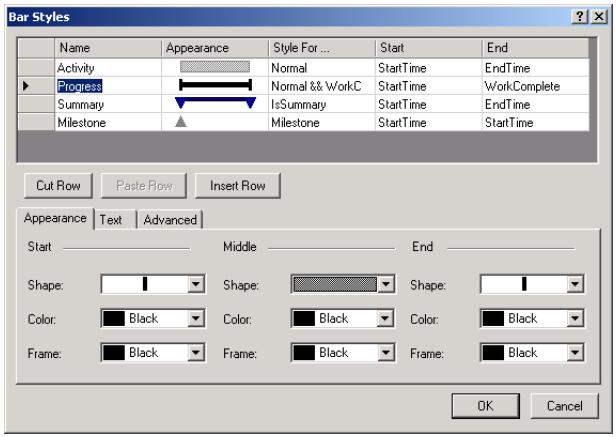
- Style 2



- Style 3



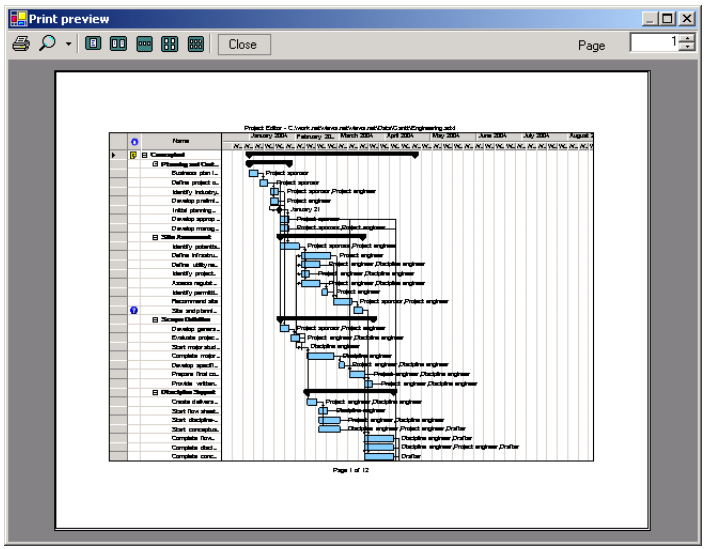
A dialog box can be used at design time and at runtime (in WinForms applications) to define the styles.



Printing

Since IBM ILOG Gantt for .NET relies on the printing services provided by the .NET framework, you can use all the services provided in the framework. You have access to the usual printing parameters such as page setting, header and footer, and to a print preview mode.

The following illustration shows the Print Preview mode.



Internationalization and Localization

All the controls support all the locales and can be used in right-to-left mode for Arabic and other languages that are written right-to-left.

Pre-built Dialogs for Windows Forms Applications

Since it is easier to edit activities, constraints or resources with text-based dialogs than with tables or Gantt sheets, IBM ILOG Gantt for .NET provides pre-built editing dialogs that can be customized.

See Also *Representing Activity Bars in Gantt Sheets* | *Printing Gantt Charts* | *Localizing a Gantt Application* | *Using Predefined Dialog Boxes for Editing Scheduling Data*

Document Conventions

The following table shows the typographic conventions used in the IBM® ILOG® Gantt for .NET documentation.

Convention	Description	Example
Monospace	Indicates code lines embedded in text and code examples.	`Public Class`
italic	Indicates the placeholders that represent the information supplied by the implementation or the user.	*context* parameter
Bold	Indicates most predefined programming elements, including namespaces, classes, delegates, objects, interfaces, methods, functions, macros, structures, constructors, properties, events, enumerations, fields, operators, statements, directives, data types, keywords, exceptions, non-HTML attributes, and configuration tags, as well as registry keys, subkeys, and values. Also indicates the following HTML elements: attributes, directives, keywords, values, and headers.	**Path** class **Resolve** method

Convention	Description	Example
Capital letters	Indicates the names of keys and key sequences.	ENTER CTRL+R
Plus sign	Indicates a combination of keys. For example, ALT+F1 means to hold down the ALT key while pressing the F1 key.	ALT+F1

System Requirements

To ensure adequate performance, the IBM® ILOG® Gantt for .NET has the following minimum and recommended system requirements.

.NET Framework Requirements

IBM ILOG Gantt for .NET 4.0 Windows® Forms and Web Forms components require the .NET Framework, version 2.0.

To build AJAX enabled applications, IBM ILOG Gantt for .NET requires by default the .NET Framework 3.5 SDK. In case you want to target a .NET Framework 2.0 Web application and ASP.NET AJAX 1.0,IBM ILOG Gantt for .NET provides the **ILOG.Gantt.Web.Ajax10.dll** assembly which has a dependency with ASP.NET AJAX 1.0. Note that in this case, the IBM ILOG Gantt for .NET Project Templates cannot be used and the Web controls must be added manually to the Visual Studio Toolbox.

To build Silverlight applications with IBM ILOG Gantt for .NET you need to have the Silverlight v2.0 installed.

Please refer to the Microsoft® documentation for operating systems and hardware requirements for each Microsoft® .NET framework version.

Visual Studio Requirements

IBM ILOG Gantt for .NET 4.0 integrates into Visual Studio 2008.

It is possible to develop IBM ILOG Gantt for .NET applications without having Visual Studio 2008 installed, but you will not take advantage of the Visual Studio integration features, like for example documentation, toolbox integration, and so on.

To build Silverlight applications inside Visual Studio 2008, you will need to install the Microsoft® Silverlight tools for Visual Studio 2008.

License Management

In order to compile or run applications using the IBM® ILOG® Gantt for .NET Framework, you need a valid license.

You might have received a license file whose content looks like:

```
LICENSE MyCompany
NODE     Gantt.NET  3.500 NEVER BSHS80YYVHG c85437cb
RUNTIME  Gantt.NET  3.500 NEVER FWEFW5345KK
```

Your license must be installed before you can start developing applications using IBM ILOG Gantt for .NET. You can install a license for the product in two different ways:

- The license can be specified during the installation process of IBM ILOG Gantt for .NET. Before the installation completes, a dialog asks you to register the license. If you accept, the **IBM ILOG License Manager** tool is launched.
- The license can be installed after the installation process by using the **IBM ILOG License Manager**.

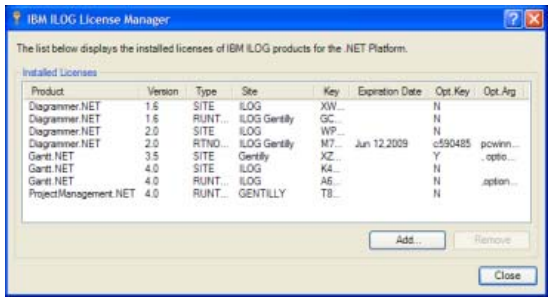
The **IBM ILOG License Manager** tool is available from the Windows® **Start Menu** in **All Programs>IBM ILOG>IBM ILOG Gantt for .NET>License Manager.**

You can use the **IBM ILOG License Manager** tool to:

- List installed licenses.
- Add new licenses.

- Remove existing licenses.

The following picture shows the **IBM ILOG License Manager**:



When compiling an application using IBM ILOG Gantt for .NET under Visual Studio 2008, the .NET license compiler will embed your license inside the final application executable file.

Since the license is embedded in the final application, there is no need to install any license on the target machine when deploying a compiled application. For the same reason, you need to recompile your application if your license has changed. For example, if you have started using IBM ILOG Gantt for .NET using an Evaluation license (the second line of your license starting with **EVAL**), the evaluation license is also embedded in the compiled application. Thus, the compiled application will stop working when the evaluation period is over. To move from an evaluation license to a normal license you need to install the new license using the **IBM ILOG License Manager** tool. Then you need to rebuild the application so that the new license will be embedded in the final application. Note that if you are using Visual Studio 2008, you must use the Rebuild command to get the application resources recompiled.

The process of compiling the license inside the final application is automatically done by Visual Studio 2008: when a licensed component is dropped onto a design surface, a license file is automatically created and added to the project. This license file contains the name of the licensed components used by the project. When building the project, Visual Studio 2008 will compile the license file and put the compiled license into the application resources.

This means that:

- If you are not using Visual Studio 2008 to compile your application, you will have to compile a license by yourself using the .NET license compiler (`lc.exe`). Then, you will have to embed the compiled license into the application resources.
- If you are using Visual Studio 2008 but you did not use the toolbox to drag and drop licensed components, you should create a `license.licx` file into your project. This file will be automatically compiled into the resources when building the project. Each line of the file should represent the fully qualified name of a licensed component.

Support

IBM Software Support Handbook

This guide contains important information on the procedures and practices followed in the service and support of your IBM products. It does not replace the contractual terms and conditions under which you acquired specific IBM Products or Services. Please review it carefully. You may want to bookmark the site so you can refer back as required to the latest information. The "IBM Software Support Handbook" can be found on the web at

http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html

Accessing Software Support

When calling or submitting a problem to IBM Software Support about a particular service request, please have the following information ready:

IBM Customer Number

The machine type/model/serial number (for Subscription and Support calls)

Company name

Contact name

Preferred means of contact (voice or email)

Telephone number where you can be reached if request is voice

Related product and version information

Related operating system and database information

Detailed description of the issue

Severity of the issue in relationship to the impact of it affecting your business needs

Contact via Web

Open service requests is a tool to help clients find the right place to open any problem, hardware or software, in any country where IBM does business. This is the starting place when it is not evident where to go to open a service request.

Service Request (SR) tool offers Passport Advantage clients for distributed platforms online problem management to open, edit and track open and closed PMRs by customer number. Timesaving options: create new PMRs with prefilled demographic fields; describe problems yourself and choose severity; submit PMRs directly to correct support queue; attach troubleshooting files directly to PMR; receive alerts when IBM updates PMR; view reports on open and closed PMRs.

You can find information about assistance for SR at http://www.ibm.com/software/support/help-contactus.html.

System Service Request (SSR) tool is similar to Electronic Service request in providing online problem management capability for clients with support offerings in place on System i, System p, System z, TotalStorage products, Linux, Windows, Dynix/PTX, Retail, OS/2, Isogon, Candle on OS/390 and Consul z/OS legacy products.

IBMLink - SoftwareXcel support contracts offer clients on the System z platform the IBMLink online problem management tool to open problem records and ask usage questions on System z software products. You can open, track, update, and close a defect or problem record; order corrective/preventive/toleration maintenance; search for known problems or technical support information: track applicable problem reports: receive alerts on high impact problems and fixes in error; and view planning information for new releases and preventive maintenance.

Contact via Phone

If you have an active service contract maintenance agreement with IBM , or are covered by Program Services, you may contact customer support teams via telephone. For individual countries, please visit the Technical Support section of the IBM Directory of worldwide contacts via http://www.ibm.com/planetwide/.

Index