
ComponentOne

GanttView for WinForms

Copyright © 1987-2013 GrapeCity, Inc. All rights reserved.

ComponentOne, a division of GrapeCity

201 South Highland Avenue, Third Floor

Pittsburgh, PA 15206 • USA

Internet: info@ComponentOne.com

Web site: <http://www.componentone.com>

Sales

E-mail: sales@componentone.com

Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Trademarks

The ComponentOne product name is a trademark and ComponentOne is a registered trademark of GrapeCity, Inc. All other trademarks used herein are the properties of their respective owners.

Warranty

ComponentOne warrants that the original CD (or diskettes) are free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective CD (or disk) to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for a defective CD (or disk) by sending it and a check for \$25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original CD (or disks) set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. We are not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only.

This manual was produced using [ComponentOne Doc-To-Help™](#).

Table of Contents

ComponentOne GanttView for WinForms Overview	1
Help with ComponentOne Studio for WinForms.....	1
GanttView for WinForms Key Features.....	3
GanttView for WinForms Quick Start.....	7
Step 1 of 4: Create a New Project.....	7
Step 2 of 4: Add Tasks to the Project	7
Step 3 of 4: Add Resources, Constraints, and Predecessors to the Tasks	9
Step 4 of 4: Styling the Task Bars	10
Design-Time Support.....	11
C1GanttView Context Menu.....	11
C1GanttView Smart Tag.....	13
C1GanttView Collection Editors.....	14
BarStyles Collection Editor.....	14
Columns Collection Editor.....	15
CustomCalendars Collection Editor	16
CustomCalendar CalendarException Collection Editor	16
CustomCalendar WorkWeek Collection Editor	17
Resources Collection Editor	18
Task Collection Editor	19
Run-Time Support.....	20
C1GanttView Dialog Boxes.....	20
Bar Styles Dialog Box	20
Change Working Time Dialog Box	21
Grid Columns Dialog Box.....	25
Progress Line Dialog Box	25
Project Information Dialog Box.....	26
Project Resources Dialog Box	28
Task Information Dialog Box.....	30
Timescale Dialog Box	35
C1GanttView Toolbar.....	36

Timescale Formats	37
GanttView Appearance.....	40
C1GanttView Appearance Properties	40
CellBorderColor Example	41
FixedBackColor Example.....	41
FixedCellBorderColor Example.....	41
FixedForeColor Example	41
HighlightBackColor Example.....	42
HighlightForeColor Example.....	42
NonworkingTimeColor Example	42
SplitterColor Example.....	43
StartFinishLineColor Example.....	43
TodayLineColor Example	43
ToolbarBackColor Example	44
Optional Elements	44
Visual Styles	44
Bar Styles.....	46
Task Elements.....	46
Task Mode.....	47
Manual Tasks.....	47
Automatic Tasks.....	47
Changing Task Scheduling Modes	47
Task Constraints	47
Task Deadline	48
Task Duration	48
Task Notes.....	48
Task Predecessor.....	49
Task Start and Finish Time.....	50
Task Percent Complete.....	51
Task Resources.....	51
Milestones	52
Progress Lines	52
GanttView for WinForms Samples	53
GanttView for WinForms Task-Based Help	55
Adding Columns to the Grid	55
Moving Columns in the Grid.....	55

Adding a Note to the Task.....	56
Assigning Resources to a Task.....	57
Setting a Tier for the TimeScale	59
Deleting a Task	59
Inserting a Task.....	60
Moving a Task	62
Creating a Milestone.....	63
Creating Predecessors.....	64
Adding a Predecessor Programatically	64
Creating a Finish to Start Predecessor Type.....	65
Creating a Start to Start Predecessor Type	66
Creating a Finish to Finish Predecessor Type.....	67
Creating a Start to Finish Predecessor Type.....	68
Adding a Vacation Day for a Resource.....	70
Saving and Loading GanttView as an XML File.....	70
Loading GanttView From an XML File	70
Saving GanttView as an XML File.....	72
Customizing the Bar Style.....	73
Creating a Custom Column	74
Showing the Duration Columns in the Grid	75
Changing the Month the Fiscal Year Starts On	76
Modifying the Progress Lines in your Project	76

ComponentOne GanttView for WinForms Overview

The C1GanttView control delivers a Microsoft Project®-like user experience for project management. It provides a graphical diagram of a schedule that helps to plan, coordinate, and track specific tasks in a project. Manage your projects effectively and efficiently with added support for constraints, dependencies, resources, styles and more.

The C1GanttView control can automatically generate a schedule from a list of tasks, durations and constraints. Or users can edit task information manually through a tabular data grid, input dialogs, or by clicking and dragging bars within the chart pane.

The C1GanttView control extends the popular C1FlexGrid control. It consists of two grids, one on the left for traditional data entry, and one on the right (known as the “chart”) which has click and drag input. The two grids are separated by a resize-able splitter. A built-in toolbar with common commands is also provided.



Getting Started

- [GanttView for WinForms Quick Start](#) (page 7)
- [GanttView for WinForms Key Features](#) (page 3)
- [GanttView for WinForms Task-Based Help](#) (page 55)

Help with ComponentOne Studio for WinForms

Getting Started

For information on installing ComponentOne Studio for WinForms, licensing, technical support, namespaces and creating a project with the control, please visit [Getting Started with Studio for WinForms](#).

What's New

For a list of the latest features added to **ComponentOne Studio for WinForms**, visit [What's New in Studio for WinForms](#).

GanttView for WinForms Key Features

The following are some of the main features of C1GanttView that you may find useful:

Automatic and Manual Scheduling

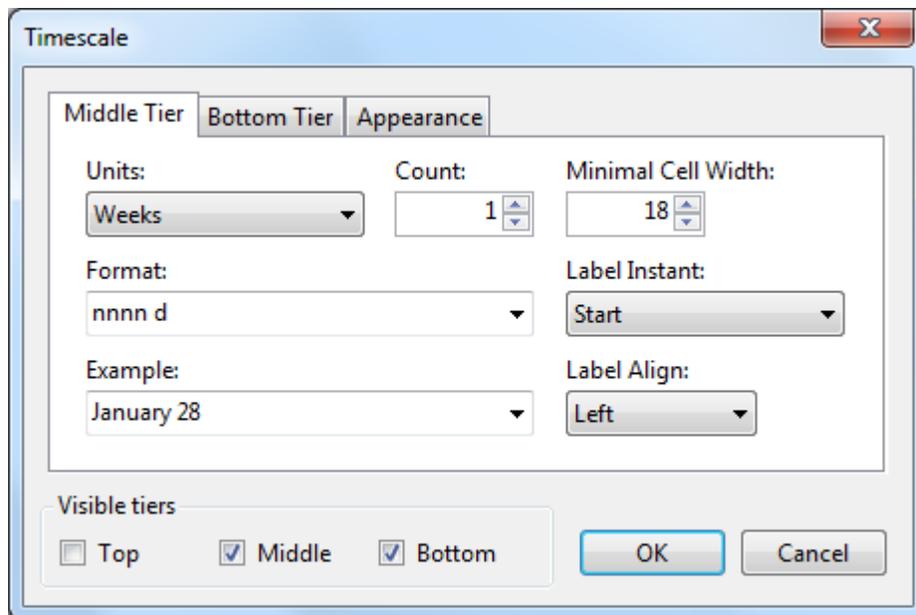
The C1GanttView control supports both Automatic and Manual scheduling modes like Microsoft Project. Automatic scheduling provides a highly structured, systematic means of managing project schedules. The project can be scheduled either from start or from the finish date. When you enter a start date, C1GanttView schedules the first auto-scheduled task to begin on that date, and calculates the sequence of auto-scheduled tasks that follow. If you enter the finish date, C1GanttView schedules the final task back from that date, and then it schedules the task before the final task, and so on, until the first task is scheduled and the project start date is calculated. C1GanttView can calculate the earliest or latest dates for tasks to complete the optimal schedule.

Manual scheduling is the default mode that gives the greatest flexibility and control over planning and managing the task schedule. With manual scheduling the C1GanttView control does not automatically configure start and finish dates of tasks.

For conceptual information see, [Task Mode](#) (page 47).

Calendar and Time Scale Configuration

Create and save custom calendar settings that specify working weeks, times and exceptions. GanttView defines the default working and nonworking days for scheduling both manually and auto-scheduled tasks. GanttView supports very flexible time scheduling including customizable work weeks and calendar exceptions with powerful recurrence patterns. Manage the time scale by specifying the date/time format for up to three tiers through the run-time dialogs.



Constraints

Constraints are a restriction set on the start or finish date of a task. Each task can be assigned a constraint type and a constraint date to which it must adhere when scheduling in automatic mode.

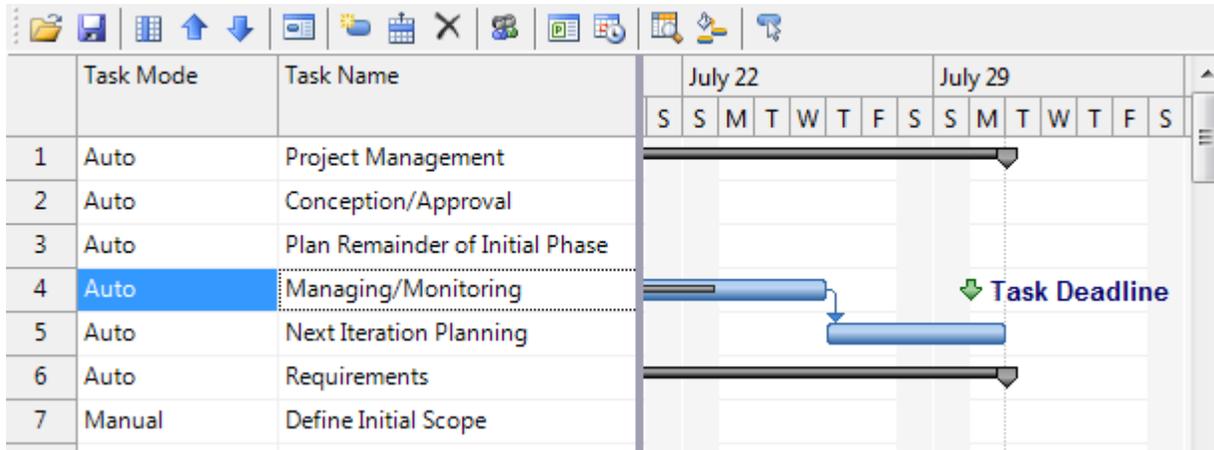
For conceptual information see, [Task Constraints](#) (page 47).

Predecessors

Each task can be assigned one or many predecessors, which are other tasks that it depends on to either start or finish. Task predecessors (or dependencies) are visualized by drawing arrows between task bars. Enter predecessors through the [Task Information dialog](#) (page 30), or by dragging a task bar onto its successor. GanttView also supports validation to alert the user when scheduling conflicts arise.

Deadlines

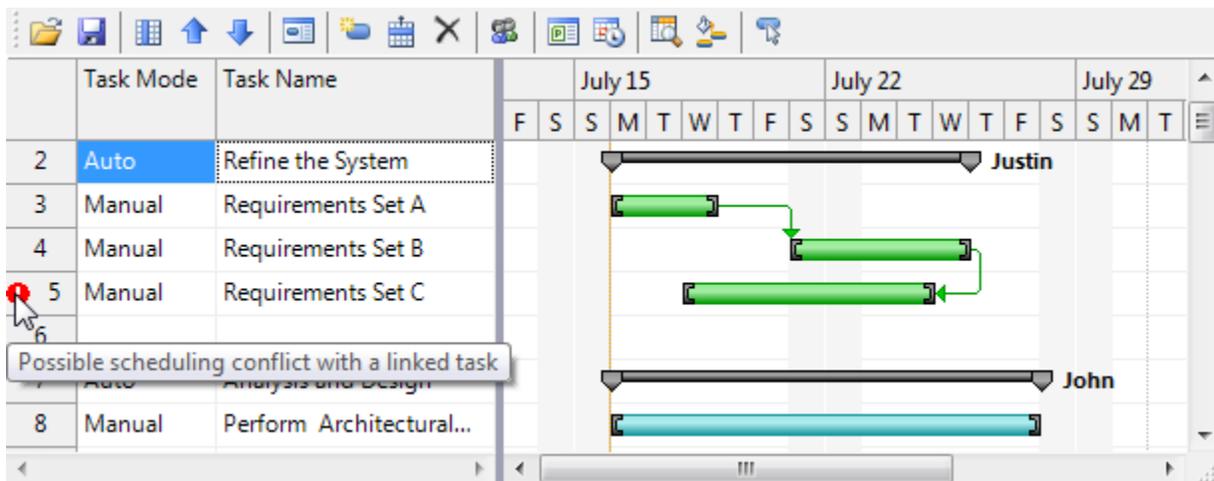
Set the deadline for any task to indicate when the task must be completed by. GanttView provides visual cues once a deadline is set. When a deadline is set for a specific task, a downward green arrow is drawn on the chart.



For conceptual information see, [Task Deadline](#) (page 48).

Conflict Validation

The C1GanttView control will alert the user when scheduling conflicts arise. There may be a conflict between a constraint and a linked task, a circular reference between tasks, or if a deadline is scheduled before a task is set to finish. A red indicator will appear next to a task containing a conflict.



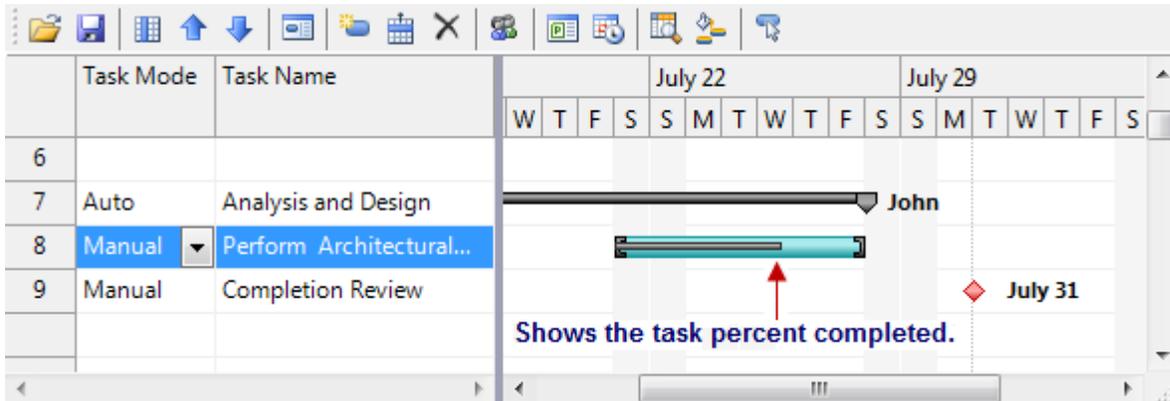
Resources

Manage a list of resources for your project, such as people, materials and costs. Assign any number of resources to a task and give them each a cost rate.

For conceptual information see, [Task Resources](#) (page 51). For procedural information see, [Assigning Resources to a Task](#) (page 57).

Percent Complete

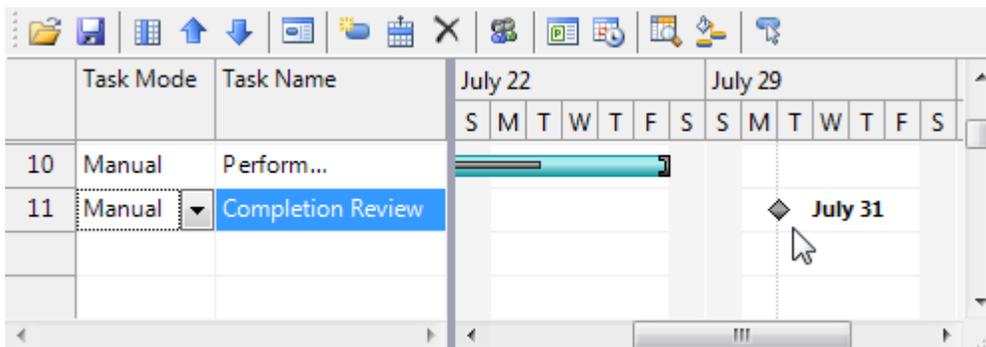
The task bars of the C1GanttView control can display like a progress bar to visualize task percentage complete.



For more information see, [Task Percent Complete](#) (page 51).

Milestones

A milestone is a significant point or landmark in your project. Milestones are created as tasks with zero duration and are visualized with a diamond shape.



Load and Save as XML

Store your project schedule as XML. The built-in toolbar includes commands for saving and loading schedules, or you can call the SaveXML and LoadXML methods in code.

For more information on using the Save and Load XML feature see, [Saving and Loading GanttView as an XML File](#) (page 70).

Custom Columns

With the C1GanttView control you can create custom columns in the grid that can save and load data from the stored XML. Each custom column has its own data type, caption, data format, and text align.

GanttView for WinForms Quick Start

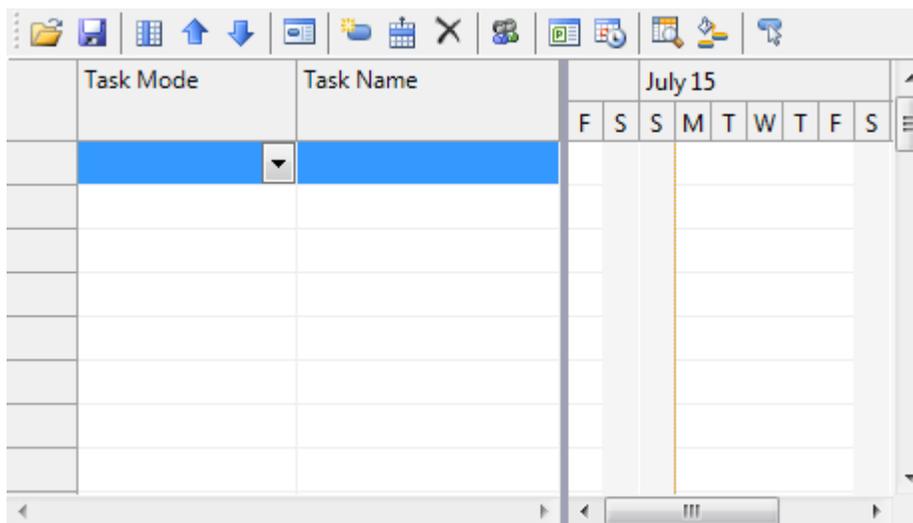
The goal of this quick start guide is to get you acquainted with **GanttView for WinForms**. In the first step of this Quick Start guide, you will add a C1GanttView control to your WinForms project. This quick start guide will also explain how to add tasks, resources, and other schedule information to the project.

Step 1 of 4: Create a New Project

In this step, you will create a .NET project and add a C1GanttView control to it.

Complete the following steps:

1. Begin by creating a new Windows Forms Application.
2. While in Design view, navigate to the Visual Studio Toolbox and double-click the C1GanttView control to add it to your form.
3. The **C1GanttView** control appears.



4. Right-click on the GanttView control and select **Properties**.
5. In the C1GanttView properties window set the **Schedule From** property to **Project Start Date** to schedule from the start date.

Note: If you need to find out how late you can begin a project set the **ScheduleFrom** to **Project Finish Date**.

6. To schedule from the start date, enter **7/2/2012** next to the **StartDate** property.

Step 2 of 4: Add Tasks to the Project

In this step you learn how to create regular tasks, milestones, and set the duration of the tasks.

To add tasks to the project at design time:

1. Right-click on the control and select **Edit Tasks**.

The **CIGanttView.Tasks Collection Editor** appears.

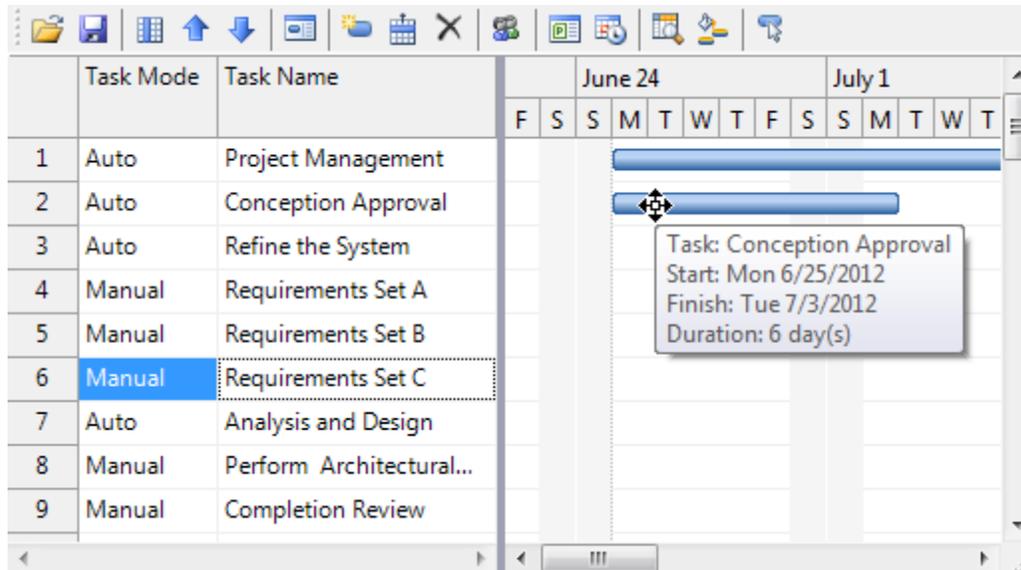
2. Select the first member from the list, member[0] and enter a name next to the **Name** property such as, **Project Management**.
3. Set the **Mode** to **Automatic**. For more information on Automatic tasks see [Task Mode](#) (page 47).
4. Set **Start** to **7/2/2012** and the **Finish** to **7/31/2012**.
5. Select the second member from the list, member[1] and enter a name next to the **Name** property such as, **Conception/Approval**.
6. Set **Start** to **7/2/2012** and **Finish** to **7/10/12**.
7. Set the **Mode** to **Automatic**.
8. Set the **PercentComplete** to **100%**. For more information see [Task Percent Complete](#) (page 51).
9. Select the third member from the list, member[2] and enter a name next to the **Name** property such as, **Requirements**.
10. Set **Start** to **7/4/2012** and **Finish** to **7/31/2012**.
11. Select the fourth member from the list, member[3] and enter a name next to the **Name** property such as, **Requirements Set A**.
12. Set **Start** to **7/18/2012** and **Finish** to **7/23/2012**.
13. Select the fifth member from the list, member[4] and enter a name next to the **Name** property such as, **Requirements Set B**.
14. Set **Start** to **7/23/2012** and **Finish** to **7/27/12**.
15. Select the sixth member from the list, member[5] and enter a name next to the **Name** property such as, **Requirements Set C**.
16. Set **Start** to **7/20/2012** and **Finish** to **7/23/12**.
17. Select the seventh member from the list, member[6] and enter a name next to the **Name** property such as, **Analysis and Design**.
18. Set **Start** to **7/10/2012** and **Finish** to **7/30/12**.
19. Set the **Mode** to **Automatic**.
20. Select the eighth member from the list, member[7] and enter a name next to the **Name** property such as, **Completion Review**.
21. Set **Start** to **7/31/2012** and **Finish** to **7/31/12**.
22. Enter 0 for the **Duration**.
23. Click **OK** to save and close the **CIGanttView.Tasks Collection Editor**.

Run and observe the following:

1. Run your project and drag the grey splitter bar over to the left to expand the chartview area.
2. Notice the Automatic scheduled tasks, **Project Management** and **Conception/Approval** have a different bar color than the remaining taskss which are Manual.
3. As you scroll along the chartview area notice the vertical orange line represents today's date.
The color can be modified using the **TodayLineColor** property. For more information see [CIGanttView Appearance Properties](#) (page 40).
4. Scroll along the chartview area and notice the two vertical dotted silver lines to indicate the project start/finish dates.

The color can be modified using the **StartFinishLineColor** property. For more information see [CIGanttView Appearance Properties](#) (page 40).

5. Hover over each task bar and notice a tooltip appears with the task's summary.



6. Locate the diamond shaped bar in the chartview area.

This task represents a milestone. Notice by default the Finish date is displayed to the right of the bar.

In the next step you will learn how to add resources, constraints, and predecessors to the tasks.

Step 3 of 4: Add Resources, Constraints, and Predecessors to the Tasks

In this step you will learn now to add resources, constraints, and predecessors to specific tasks.

To add resources to the project at design time:

1. Right-click on the control and select **Edit Resources**.
The **CIGanttView.Resources Collection Editor** appears.
2. Click the **Add** button twice to add two resources to the collection.
3. Set Resource 1 **Name** to **Tim** and Resource 2 **Name** to **John**.
4. Click **OK** to save and close the **CIGanttView.Resources Collection Editor**.

To assign resources to specific tasks at design time:

1. Right-click on the control and select **Edit Tasks**.
The **CIGanttView.Tasks Collection Editor** appears.
2. Select the **Conception Approval** task and click on the ellipsis button next to **ResourceRefs**.
The **Task.ResourceRefs Collection Editor** appears.
3. Click **Add** to add a reference to **Tim**.
4. Set the **Resource** to **Tim**.

If you wanted to assign more resources to that task you could click Add again to assign another resource.

5. Click **OK** to save and close the **Task.ResourceRefs Collection Editor**.
6. Select the **Analysis and Design** task and click on the ellipsis button next to **ResourceRefs**.
The **Task.ResourceRefs Collection Editor** appears.
7. Click **Add** to add a reference to **John**.
8. Set the **Resource** to **John**.
9. Click **OK** to save and close the **Task.ResourceRefs Collection Editor**.

For more information on resources see [Task Resources](#) (page 51).

To add predecessors to the project at design time:

1. Right-click on the control and select **Edit Tasks**.
The **C1GanttView.Tasks Collection Editor** appears.
2. Select the **Requirements Set B** task and click on the ellipsis button next to **Predecessors**.
The **Task.Predecessors Collection Editor** appears.
3. Click **Add** to add a predecessor.
4. Set the **Predecessor Task** to **Requirements Set A**.
5. Click **OK** to save and close the **Task.Predecessors Collection Editor**.
6. Select the **Requirements Set C** task and click on the ellipsis button next to **Predecessors**.
The **Task.Predecessors Collection Editor** appears.
7. Click **Add** to add a predecessor.
8. Set the **Predecessor Task** to **Requirements Set B**.
9. Set the **PredecessorType** to **FinishToStart**.
10. Click **OK** to save and close the **Task.Predecessors Collection Editor**.

For more information on predecessors see [Creating Predecessors](#) (page 64).

To add constraints to the project at design time:

1. Right-click on the control and select **Edit Tasks**.
The **C1GanttView.Tasks Collection Editor** appears.
2. Select the **Requirements** task and set the **ConstraintType** to **StartNoEarlierThan**.
3. Set the **ConstraintDate** to **7/4/2012**.
4. Select the **Analysis and Design** task and set the **ConstraintType** to **MustStartOn**.
5. Set the **ConstraintDate** to **7/10/2012**.
6. Click **OK** to save and close the **C1GanttView.Tasks Collection Editor**.

For more information on constraints see [Task Constraints](#) (page 47).

Step 4 of 4: Styling the Task Bars

In this step you will learn how to modify the style of the task bars for the automatic and manual tasks using the **C1GanttView.BarStyles Collection Editor**.

To modify the bar styles at design time:

1. Right-click on the **C1GanttView** control and select **Edit Bar Styles**.

The **C1GanttView.BarStyles Collection Editor** appears.

2. Click **Add** to add a bar style to the collection.
3. Set the **BarType** to **AutoTask**.
4. Set the **BarShape** to **ThickBar**.
5. Set the **BarColor** to **LightSkyBlue**.
6. Click **Add** to add a bar style to the collection.
7. Set the **BarType** to **ManualTask**.
8. Set the **BarShape** to **ThickBar**.
9. Set the **BarColor** to **PaleGreen**.
10. Click **OK** to save and close the **C1GanttView.BarStyles Collection Editor**.

To modify the bar styles for a specific task at design time:

1. Right-click on the control and select **Edit Tasks**.

The **C1GanttView.Tasks Collection Editor** appears.

2. Select the **Analysis and Design** task and click on the ellipsis button next to the **BarStyles**.

The **C1GanttView.BarStyles Collection Editor** appears.

3. Click **Add** to add a bar style to the collection.
4. Set the **BarType** to **AutoTask**.
5. Set the **BarShape** to **TopBar**.
6. Set the **StartShape** and **EndShape** to **2**.
7. Set **RightText2** to **ResourceNames**.

For more information see [Bar Styles](#) (page 46).

What You've Accomplished

Congratulations! You have successfully completed the C1GanttView quick start. In this topic, you added a C1GanttView control to your windows form, added tasks, customized its behavior and appearance, and manipulated the control at run time.

Design-Time Support

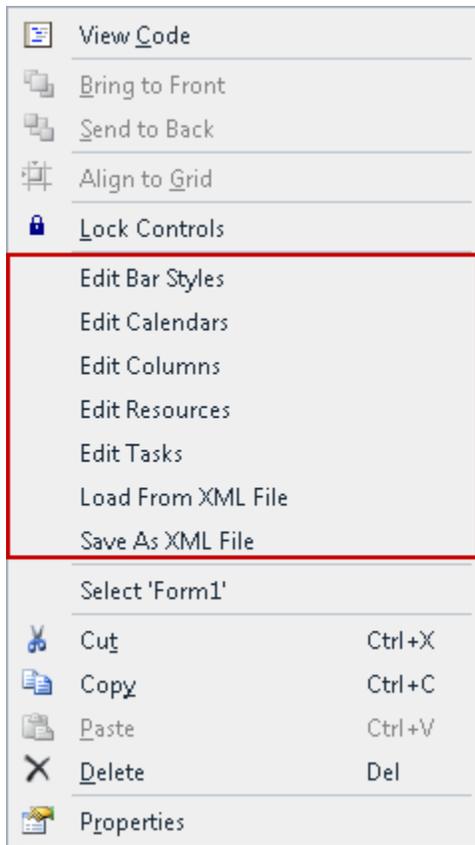
C1GanttView provides customized context menus, smart tags, and a designer that offers rich design-time support and simplifies working with the object model.

The following topics describe how to use **C1GanttView** design-time environment to configure **C1GanttView**.

C1GanttView Context Menu

The **C1GanttView** control provides a context menu for additional functionality to use at design time.

To access C1GanttView's context menu, right-click on the **C1GanttView** control and the context menu for it appears like the following:



The **C1GanttView** context menu operates as follows:

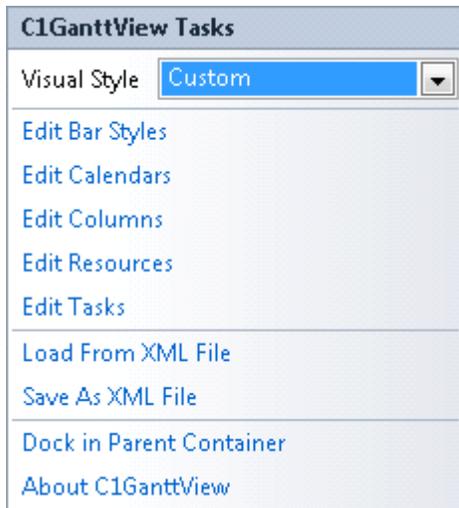
- **Edit Columns**
Selecting the **Edit Columns** opens the **C1GanttView.Columns Collection Editor** where you can add, remove, or modify task property columns and custom field columns.
- **Edit Calendars**
Selecting the **Edit Calendars** item opens the **C1GanttView.CustomCalendars Collection Editor** where you can add, remove, or modify calendars.
- **Edit Tasks**
Selecting the **Edit Tasks** item opens the **C1GanttView.Tasks Collection Editor** where you can add, remove, or modify tasks for the C1GanttView control.
- **Edit Bar Styles**
Selecting the **Edit Bar Styles** will open the **C1GanttView.BarStyles Collection Editor** where you can add, remove, or modify bar styles for the bars that represent the milestones.
- **Edit Resources**
Selecting the **Edit Resources** item will open the **C1GanttView.Resources Collection Editor** where you can add, remove, or modify the different types of resources for the ganttview..
- **Load From Xml File**
Selecting the **Load from Xml File** opens the **Load From Xml File** dialog box where you browse to the .xml file you wish to load.
- **Save As Xml File**

Selecting the **Save As Xml File** opens the **Save As Xml File** dialog box where you browse to the .xml file you wish to save.

C1GanttView Smart Tag

In Visual Studio, each component in **GanttView for WinForms** includes a smart tag. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each control.

To access the **C1GanttView Tasks** menu, click the smart tag (▾) in the upper right corner of the **C1GanttView** control. This will open the **C1GanttView Tasks** menu.



The **C1GanttView Tasks** menu operates as follows:

Visual Style

The Visual Style dropdown box provides a list of styles to choose from: Custom, System, Office2007Black, Office2007Blue, Office2007Silver, Office2010Black, Office2010Blue, Office2010Silver, and Windows7.

Edit Bar Styles

Clicking **Edit Bar Styles** will open the **C1GanttView.BarStyles Collection Editor** where you can add, remove, or modify bar styles for the bars that represent the milestones.

Edit Calendars

Clicking the **Edit Calendars** item opens the **C1GanttView.CustomCalendars Collection Editor** where you can add, remove, or modify calendars.

Edit Columns

Clicking the **Edit Columns** opens the **C1GanttView.Columns Collection Editor** where you can add, remove, or modify task property columns and custom field columns.

Edit Resources

Clicking the **Edit Resources** item will open the **C1GanttView.Resources Collection Editor** where you can add, remove, or modify the different types of resources for the ganttview.

Edit Tasks

Clicking the **Edit Tasks** item opens the **C1GanttView.Tasks Collection Editor** where you can add, remove, or modify tasks for the C1GanttView control.

Load From Xml File

Clicking the **Load from Xml File** opens the **Load From Xml File** dialog box where you browse to the .xml file you wish to load.

Save As Xml File

Clicking the **Save As Xml File** opens the **Save As Xml File** dialog box where you browse to the .xml file you wish to save.

C1GanttView Collection Editors

C1GanttView provides the following collection editors that allow you to apply properties to the Ganttviewelements at design time:

- C1GanttView.Columns Collection Editor
- C1GanttView.CustomCalendars Collection Editor
- C1GanttView.BarStyles Collection Editor
- C1GanttView.Resources Collection Editor
- CustomCalendar.Workweeks Collection Editor
- CustomCalendar.CalendarExceptions Collection Editor

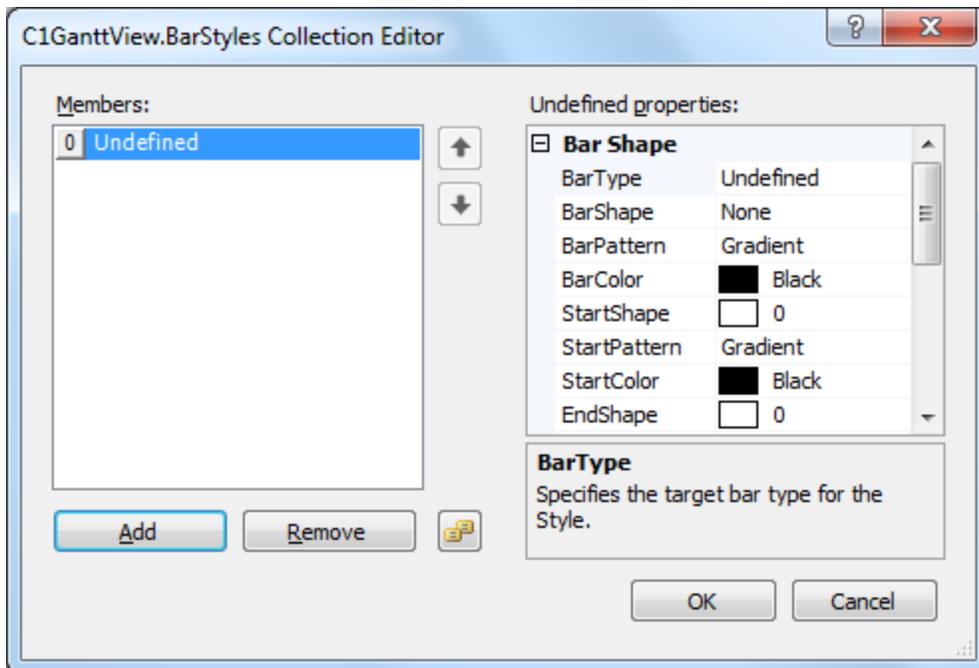
The following topics provide an overview of each GanttView collection editor and show how to access each of them:

BarStyles Collection Editor

The **C1GanttView.BarStyles Collection Editor** is used for adding different types of bars such as AutoTask, ManualTask, Progress, Milestone, Deadline, DurationOnly, StartOnly, and FinishOnly. Once the bar type is specified you can then modify its shape, pattern, color, and text.

To Access the BarStyles Collection Editor

Right-click on the **C1GanttView** control and select **Edit Bar Styles** from its context menu. The **C1GanttView.BarStyles Collection Editor** appears like the following when a member is added to the collection:

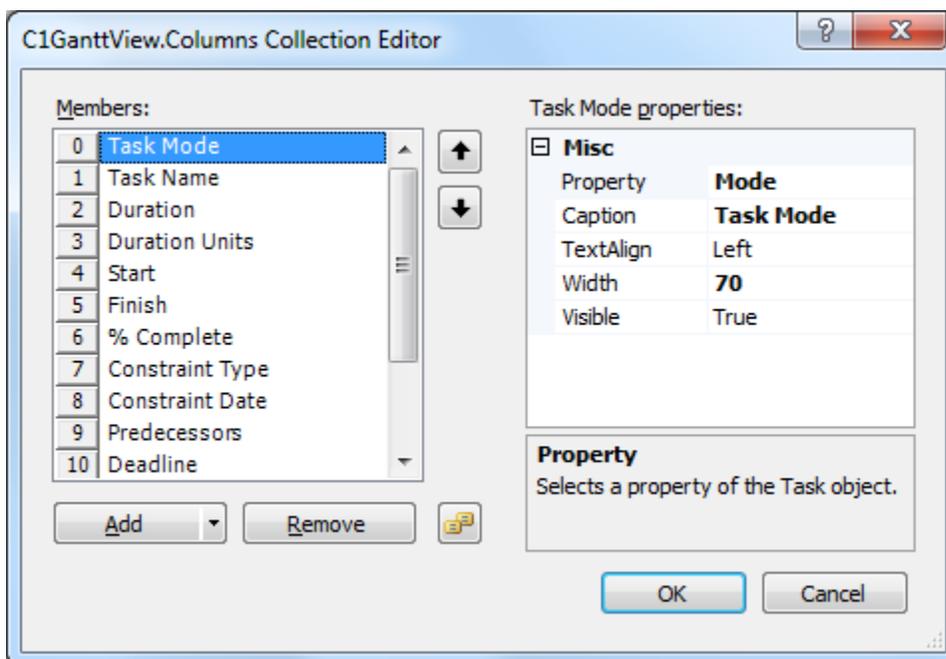


Columns Collection Editor

The **C1GanttView.Columns Collection Editor** is used for adding **TaskPropertyColumns** and **CustomFieldColumns** and then modifying its properties at design time.

To Access the Columns Collection Editor

Right-click on the **C1GanttView** control and select **Edit Columns** from its context menu. The **C1GanttView.Columns Collection Editor** appears like the following when a member is added to the collection:

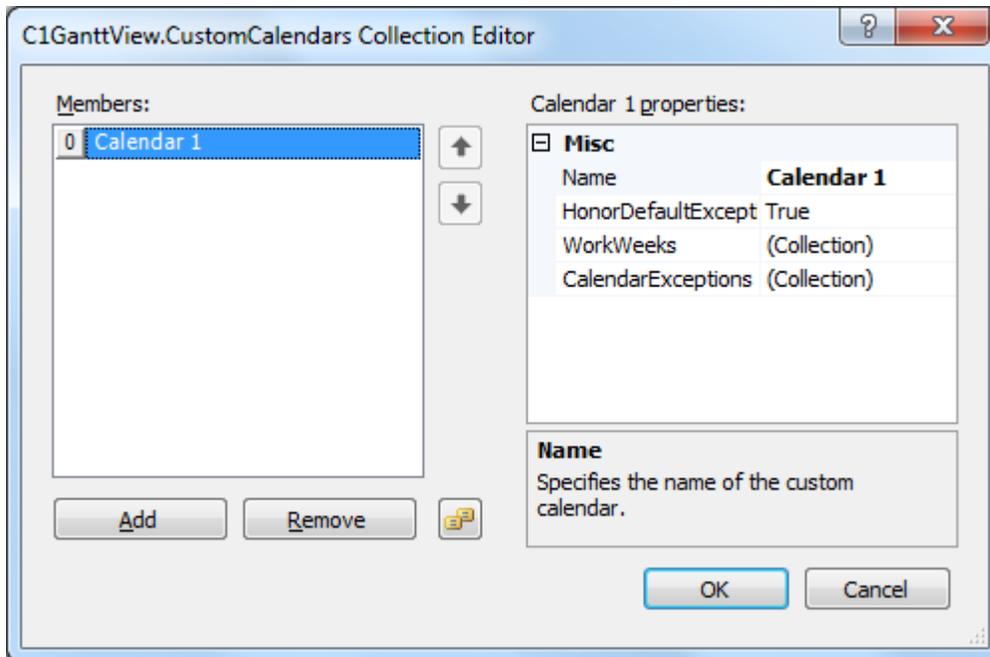


CustomCalendars Collection Editor

The **C1GanttView.CustomCalendars Collection Editor** is used for adding, removing, or modifying custom calendars at design time.

To Access the CustomCalendars Collection Editor

Right-click on the **C1GanttView** control and select **Edit Calendars** from its context menu. The **C1GanttView.CustomCalendars Collection Editor** appears like the following when a member is added to the collection:

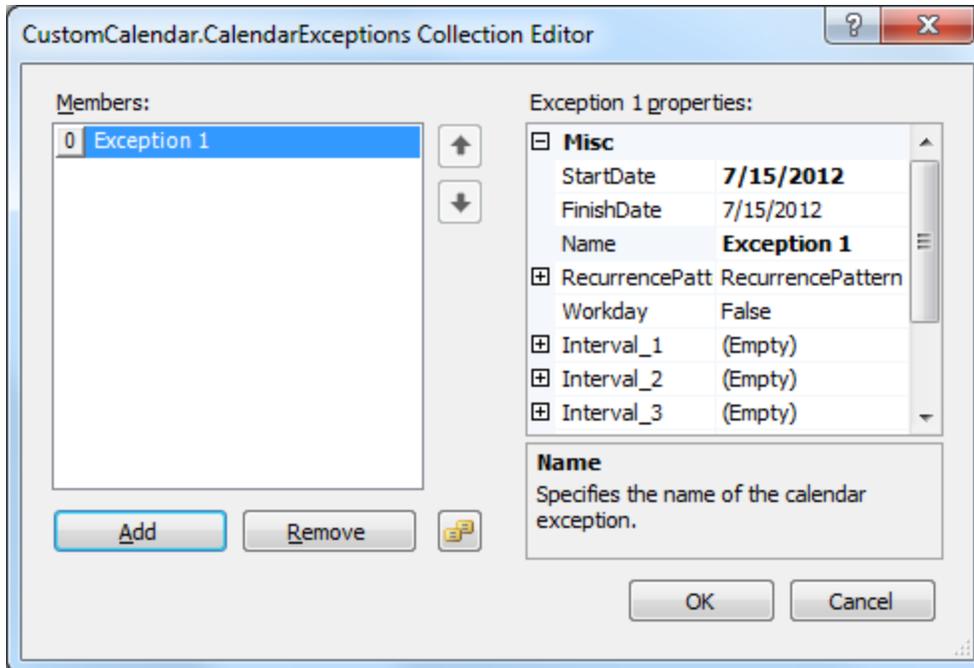


CustomCalendar CalendarException Collection Editor

The **CustomCalendar.CalendarExceptions Collection Editor** is used for adding, removing, or modifying exceptions for the custom calendars at design time.

To Access the CustomCalendar.CalendarExceptions Collection Editor

Right-click on the **C1GanttView** control and select **Edit Calendars** from its context menu. The **C1GanttView.CustomCalendars Collection Editor** appears. Click **Add** to add a new member to the collection. In the properties pane click on the ellipsis button next to **CalendarExceptions**. The **CustomCalendar.CalendarExceptions Collection Editor** appears like the following when a member is added to the collection:

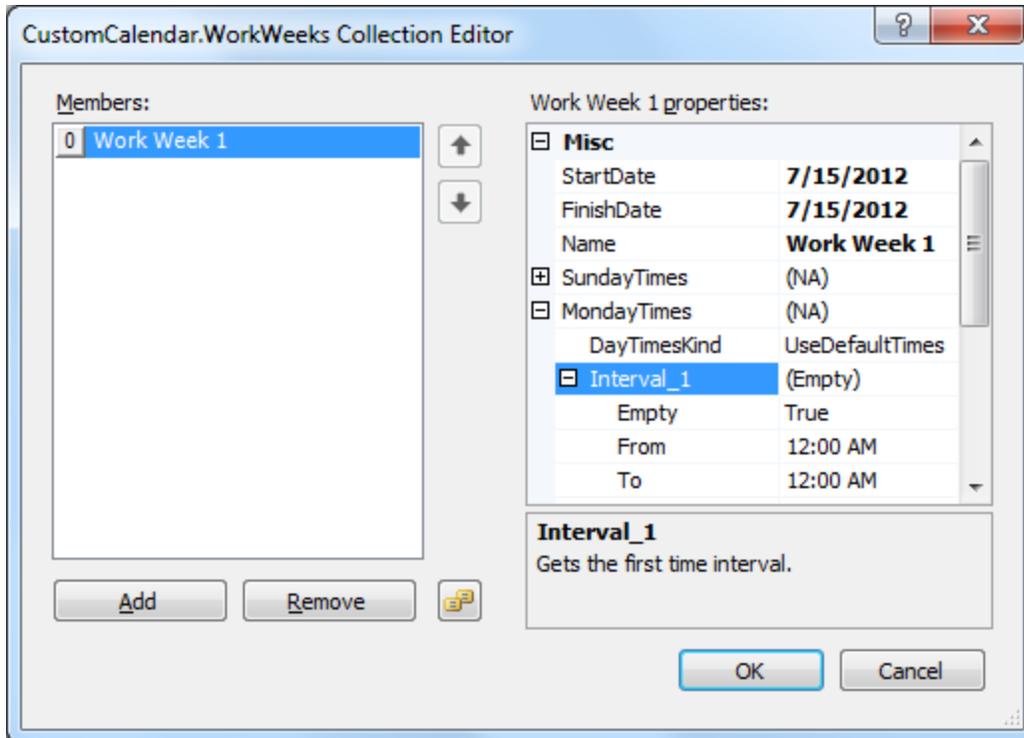


CustomCalendar WorkWeek Collection Editor

The **CustomCalendar.WorkWeeks Collection Editor** is used for adding, removing, or modifying work weeks for the custom calendars at design time.

To Access the CustomCalendar.WorkWeeks Collection Editor

Right-click on the **C1GanttView** control and select **Edit Calendars** from its context menu. The **C1GanttView.CustomCalendars Collection Editor** appears. Click **Add** to add a new member to the collection. In the properties pane click on the ellipsis button next to **WorkWeeks**. The **CustomCalendar.CalendarWorkWeeks Collection Editor** appears like the following when a member is added to the collection:

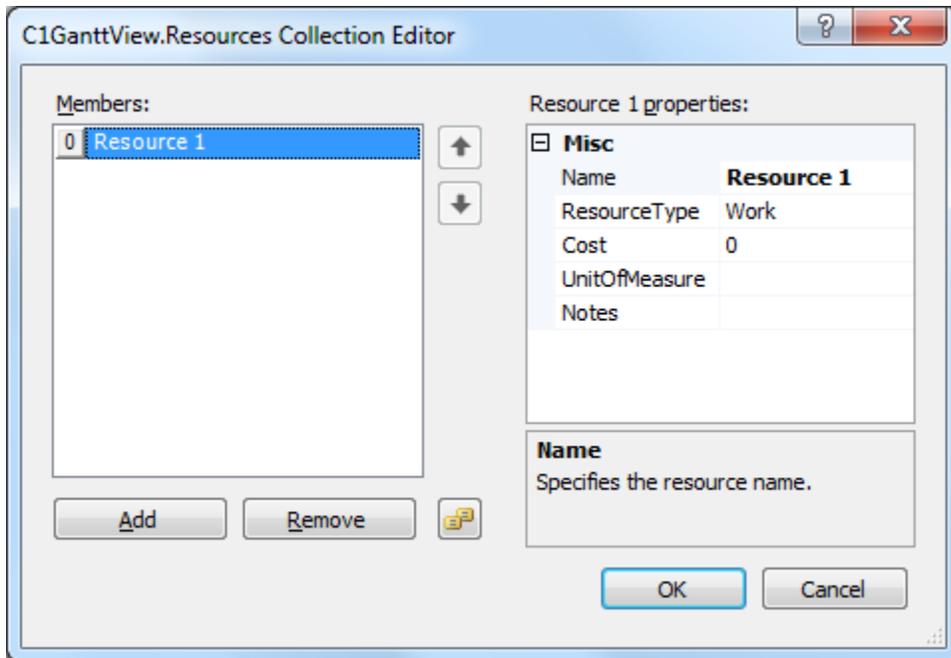


Resources Collection Editor

The **C1GanttView.Resources Collection Editor** is used for adding, removing, and modifying resources at design time.

To Access the C1GanttView.Resources Collection Editor

Right-click on the **C1GanttView** control and select **Edit Resources** from its context menu. The **C1GanttView.Resources Collection Editor** appears like the following when a member is added to the collection:

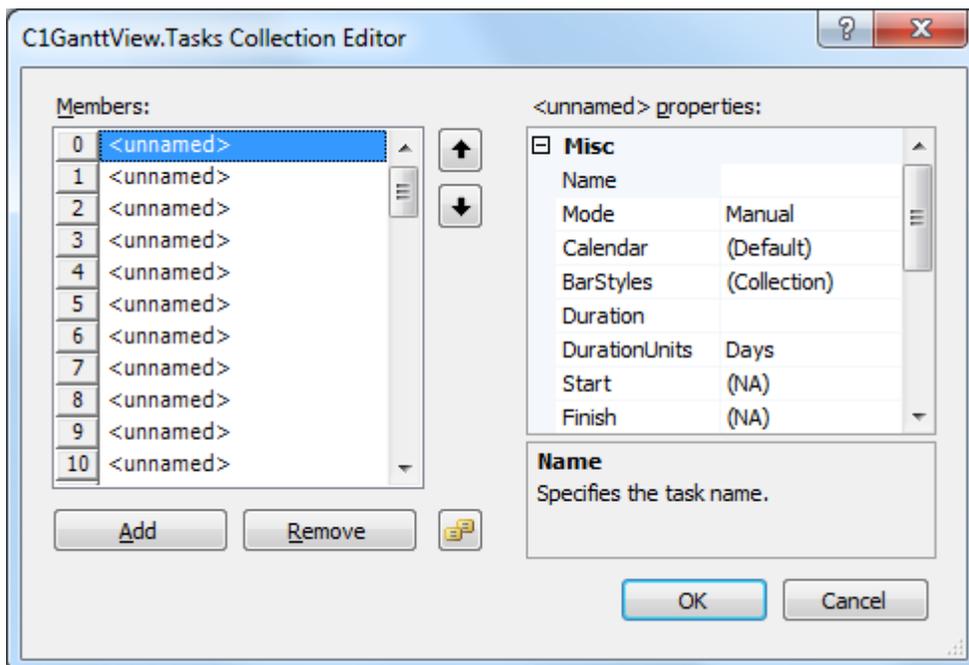


Task Collection Editor

The **C1GanttView.Tasks Collection Editor** is used for adding, removing, and modifying tasks at design time.

To Access the C1GanttView.Tasks Collection Editor

Right-click on the **C1GanttView** control and select **Edit Tasks** from its context menu. The **C1GanttView.Tasks Collection Editor** appears like the following:



Run-Time Support

The following topics describe how to use **C1GanttView** run-time environment to configure **C1GanttView**.

C1GanttView Dialog Boxes

C1GanttView provides the following dialog boxes that enable you to modify your project schedule at run time:

- [Bar Styles Dialog Box](#) (page 20)
- [Change Working Time Dialog Box](#) (page 21)
- [Grid Columns Dialog Box](#) (page 25)
- [Project Information Dialog Box](#) (page 26)
- [Project Resources Dialog Box](#) (page 28)
- [Task Information Dialog Box](#) (page 30)
- [Timescale Dialog Box](#) (page 35)

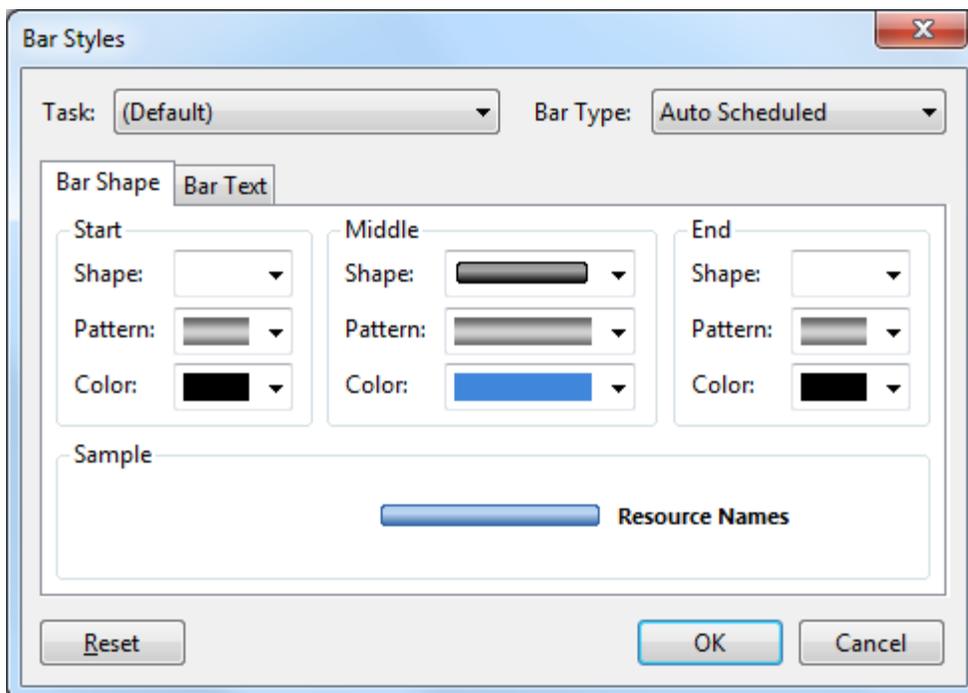
The following topics provide an overview of each C1GanttView dialog and show how to access each of them:

Bar Styles Dialog Box

The **Bar Styles** is used for creating unique styles for the different types of task bars such as AutoTask, ManualTask, Progress, Milestone, Deadline, DurationOnly, StartOnly, and FinishOnly. Once the bar type is specified you can then modify its shape, pattern, color, and text.

To Access the Bar Styles Dialog Box

Click on the **Bar Styles** button, , in the [C1GanttView Toolbar](#) (page 36).

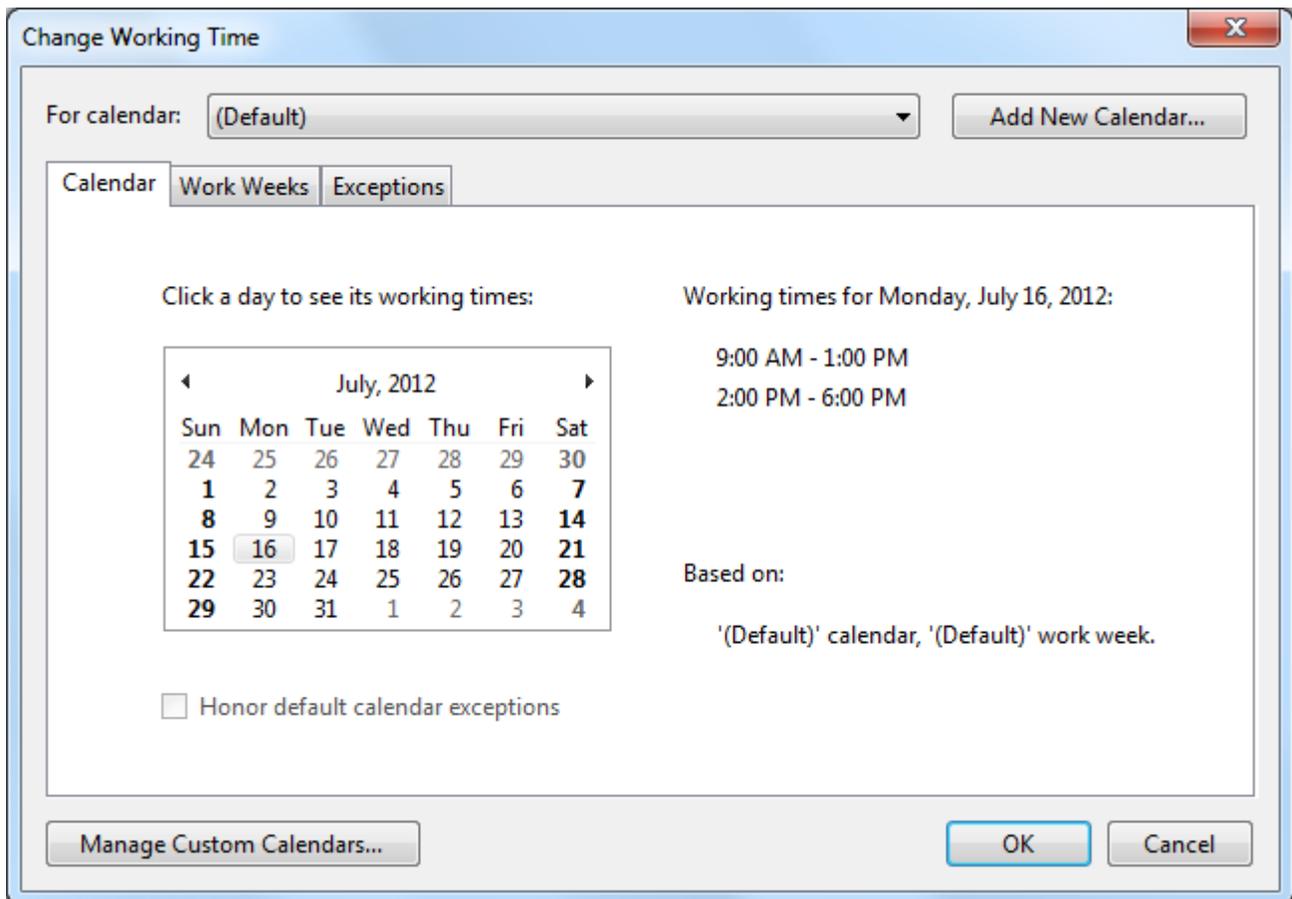


Change Working Time Dialog Box

The **Change Working Time** dialog box is used for changing the working time for the specified calendar. In the **For calendar:** dropdown listbox specify the calendar you wish to change or select **Add New Calendar** to create a new calendar or a copy of a calendar based on another calendar. This is useful when you would rather create a new calendar instead of changing the default one. Click on the **Manage Custom Calendars** if you wish to add or remove the custom calendars. Once you are finished with changing the working time in your schedule click **OK** to apply your changes to your calendar and close the **Change Working Time** dialog box.

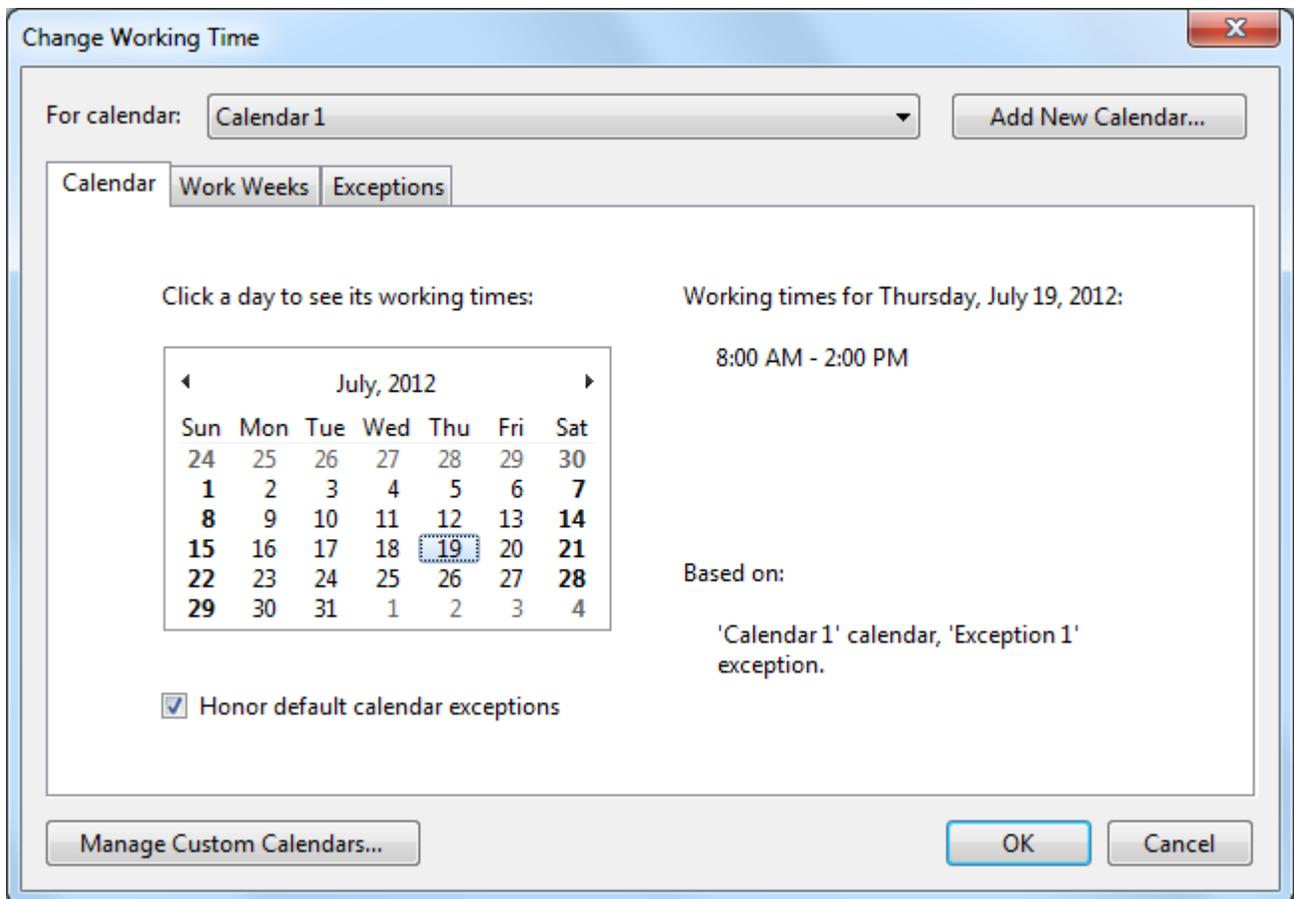
To Access the Working Time dialog box

Click on the **Change Working Time** button, , in the [C1GanttView Toolbar](#) (page 36).



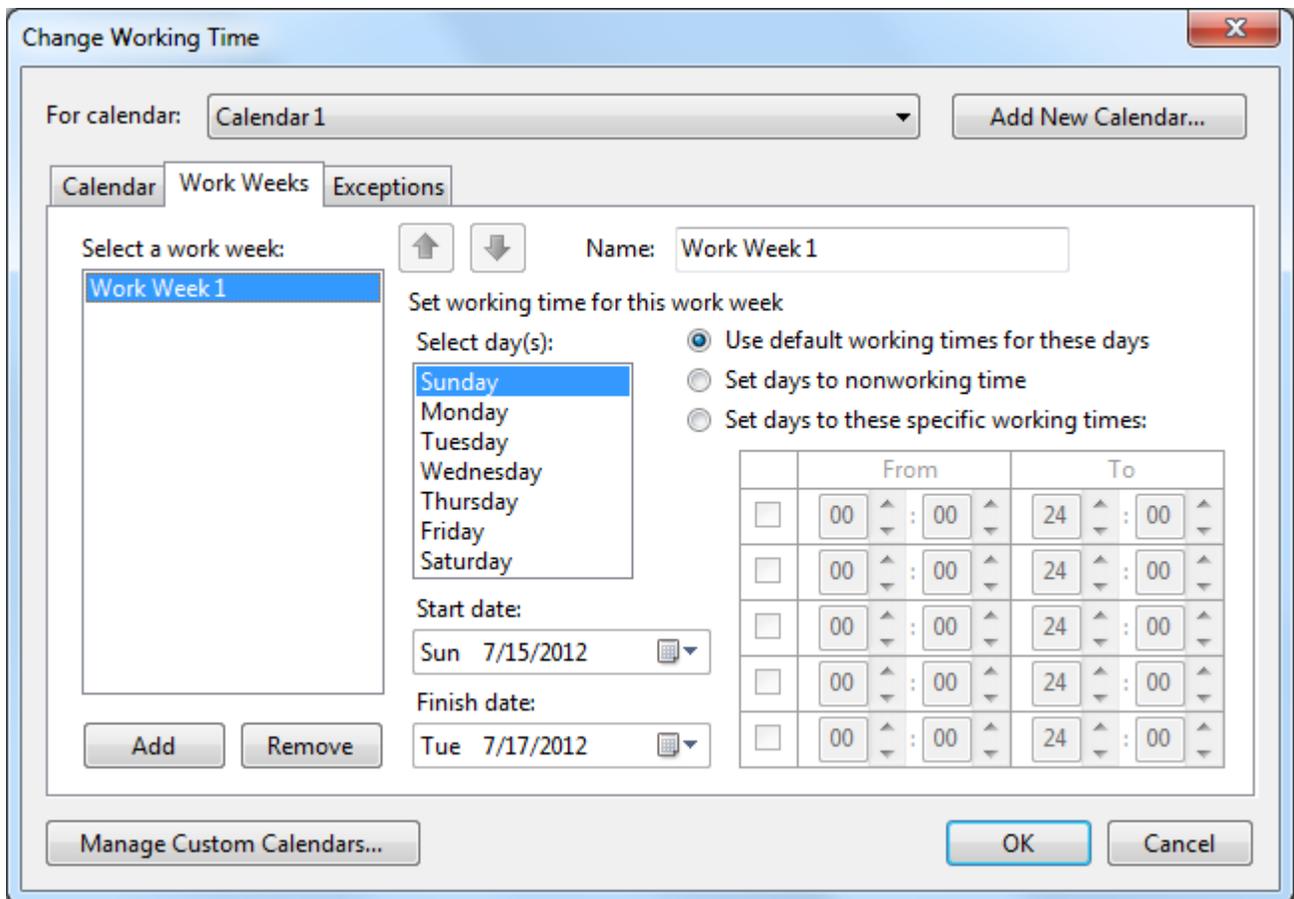
Calendar Tab

In the **Calendar** tab you can click a day in the default calendar to see its working times based on the default calendar work week and check whether or not to honor default calendar exceptions.



Work Weeks Tab

In the **Work Weeks** tab you can change the default work week for either the project calendar, resource calendar or for a new one you created. You can choose or create an additional schedule for a range of days that differ from the default work day such as work week that includes weekend days for tutoring.



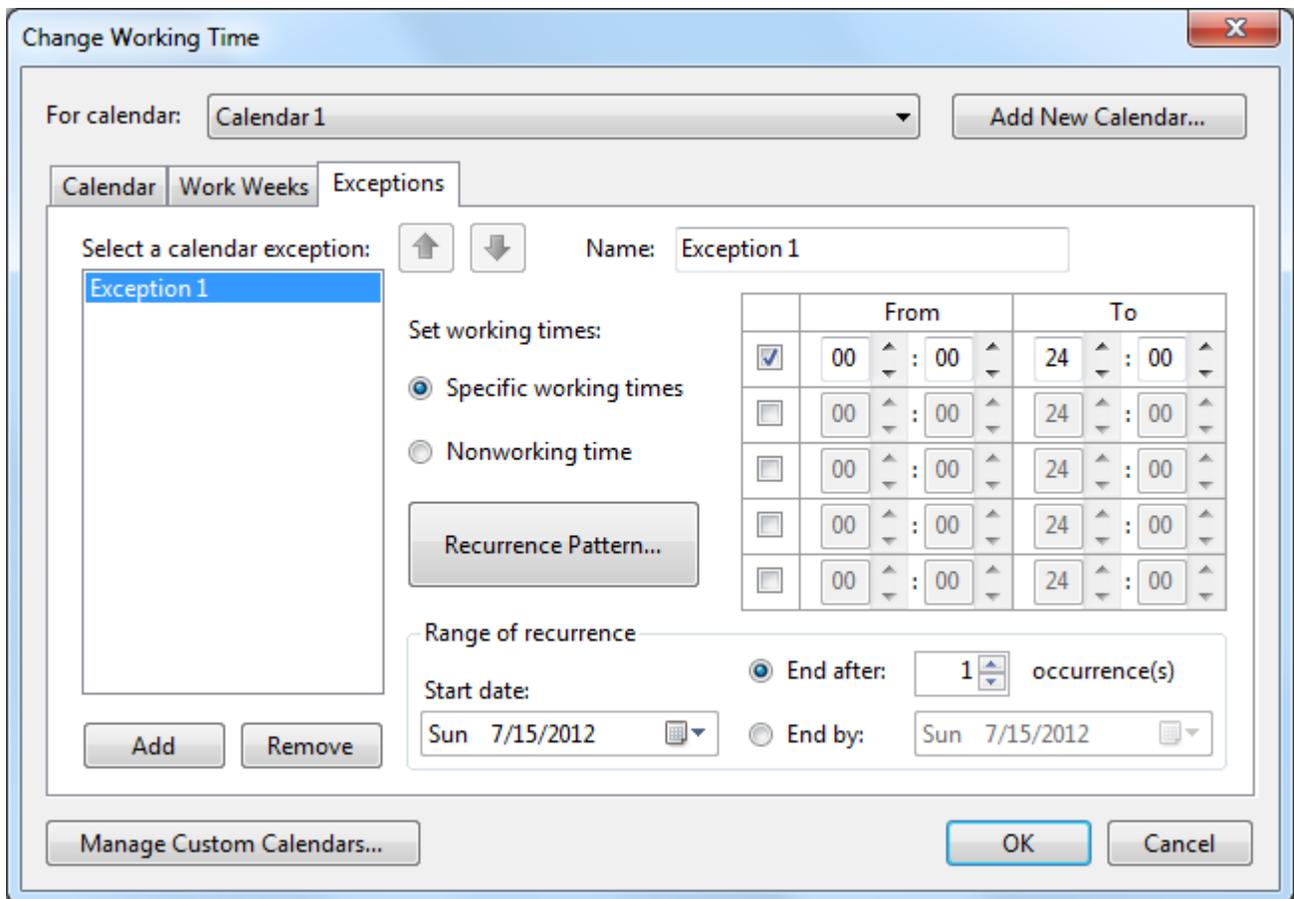
In the **Select day(s):** box, you can select one or multiple days at a time for the work week. Once you click **Add** to add a work week, type a descriptive name for the work week in the **Name:** textbox such as Summer Math Tutoring. Once the new work week is created enter the start times in the **Start date:** dropdown listbox and finish times in the **Finish date:** dropdown listbox when the additional scheduling will occur.

Once you select the day(s) you wish to change from a working day to non-working day, or vice versa, you can choose one of the following:

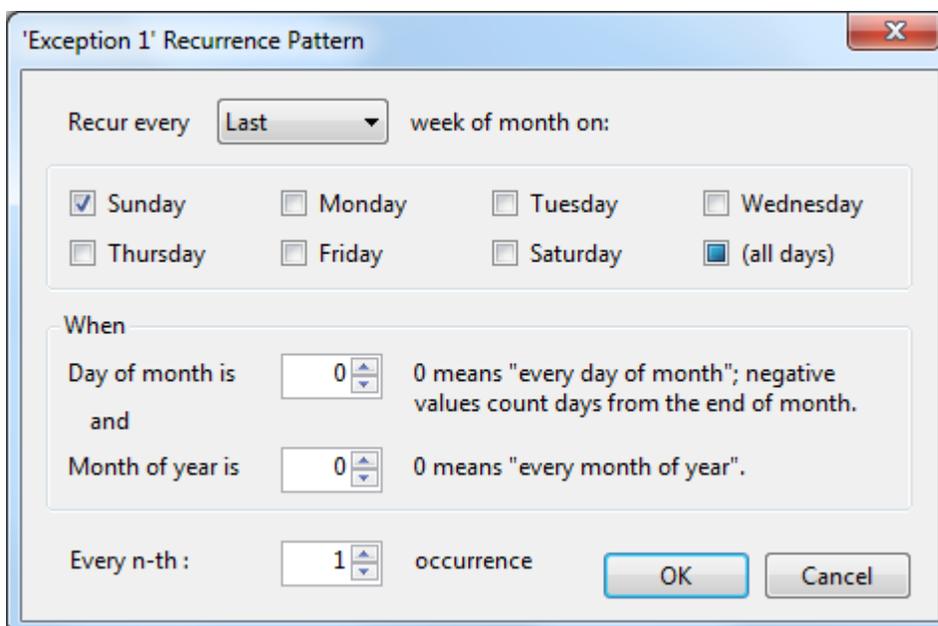
- Use Default working times for these days - Choose the days that should use the default working times, which are 8:00 A.M. to 12:00 P.M. and 1:00 P.M. to 5:00 P.M., Monday through Friday, and nonworking time on weekends.
- Set days to nonworking time - Choose the days on which no work can be scheduled. For example, if no one in your organization works on a Friday, select Friday, and then select Set days to nonworking time.
- Set days to these specific working times - To set the working times for the selected days throughout the schedule, type the times that you want work to start in the **From:** boxes and the times you want work to end in the **To:** boxes. For example, if people in your organization work on Saturdays, select Saturday, and then select the Set day(s) to these specific working times.

Exceptions Tab

In the **Exceptions** tab, click the **Add** button to add a calendar exception and type a descriptive name, such as Company Holiday, in the **Name:** textbox. Enter the start and finish times in the **Start date:** dropdown listbox and the **End by:** dropdown listbox for the time during which the exception will occur.



Click the **Recurrence Pattern...** button if the exception will be repeated throughout the section of the schedule.



In the **Recurrence Pattern** dialog box, specify the first, second, third, fourth, or last item in the **Recur every** dropdown listbox.

Select the days that the recurrence will occur, when the day of month and month of year the recurrence will occur, and how many times it will occur.

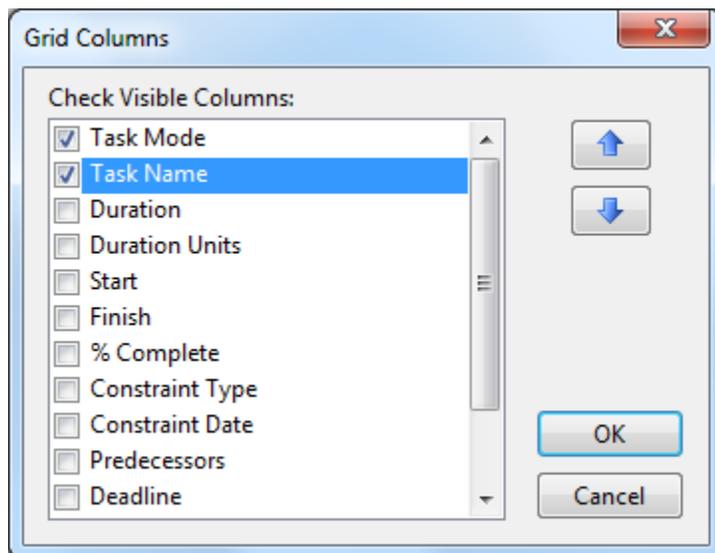
Any tasks scheduled around the calendar exception will automatically be rescheduled to take into account the nonworking time of the exception.

Grid Columns Dialog Box

The **Grid Columns dialog box** is used for specifying which attribute names appear in the columns of the **CI GanttView** control. The default visible columns are **Task Mode** and **Task Name**. Click **OK** to save and close the changes made to the **Grid Columns** dialog box.

To Access the Grid Columns dialog box

Click on the **Grid Columns** button, , in the [CI GanttView Toolbar](#) (page 36).

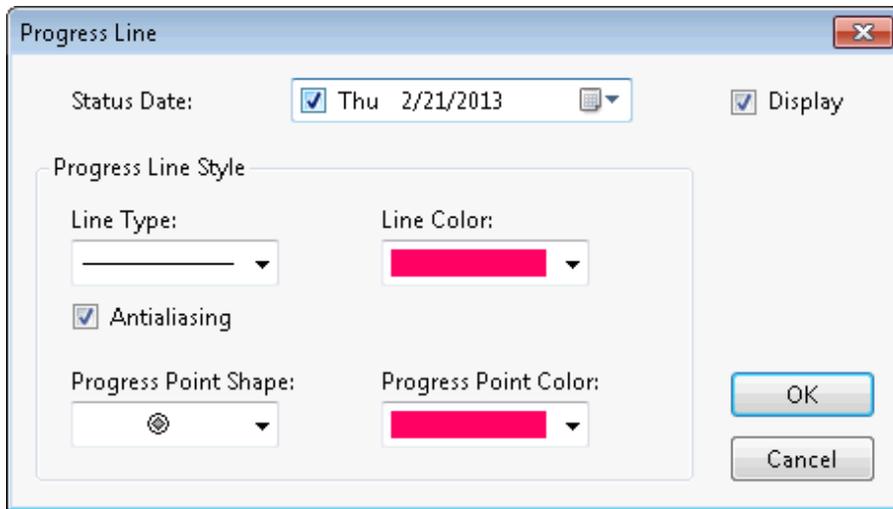


Progress Line Dialog Box

The **Progress Line** dialog box is used for displaying lines on the timescale. The progress lines draw attention to tasks that are ahead or behind the schedule.

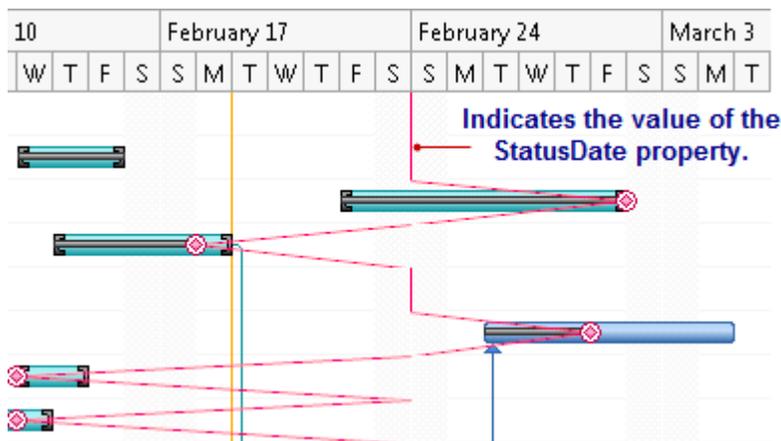
To Access the Progress Line dialog box

Click on the **Progress Line** button, , in the [CI GanttView Toolbar](#) (page 36).



The **Progress Line** tab consists of the following items:

- **Status Date:** – Click on the dropdown arrow to select the date that you wish the first start of the line to appear. For example the following shows the line starts on **Sunday February 24, 2013**.



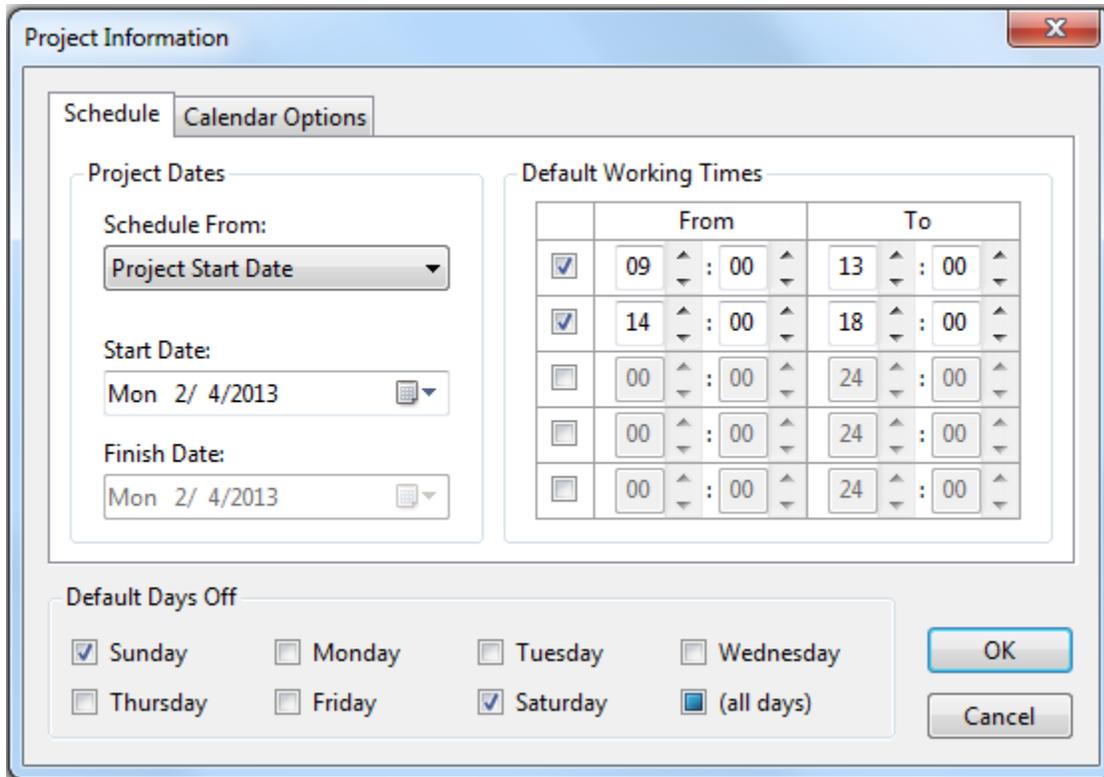
- **Show Progress Line** – Select this checkbox if you want to the progress line to appear.
- **Line Type:** – Indicates the type of line to draw for the progress line.
- **Line Color** – Specifies the color of the progress line.
- **Progress Point Shape:** – Specifies the type of shape used on the progress line.
- **Progress Point Color:** – Specifies the color of the progress point used on the progress line.

Project Information Dialog Box

The **Project Information dialog box** is used for creating and scheduling a new project. The **Project Information** dialog box consists of a Schedule tab and a Calendar Options tab. The Schedule tab includes a **Project dates** groupbox and a **Default working times** groupbox.

To Access the Project Information dialog box

Click on the **Project Information** button, , in the [C1GanttView Toolbar](#) (page 36).



Project Information

Schedule | **Calendar Options**

Project Dates

Schedule From:
Project Start Date

Start Date:
Mon 2/ 4/2013

Finish Date:
Mon 2/ 4/2013

Default Working Times

	From	To
<input checked="" type="checkbox"/>	09 : 00	13 : 00
<input checked="" type="checkbox"/>	14 : 00	18 : 00
<input type="checkbox"/>	00 : 00	24 : 00
<input type="checkbox"/>	00 : 00	24 : 00
<input type="checkbox"/>	00 : 00	24 : 00

Default Days Off

Sunday Monday Tuesday Wednesday
 Thursday Friday Saturday (all days)

OK
Cancel

The Project dates groupbox consists of:

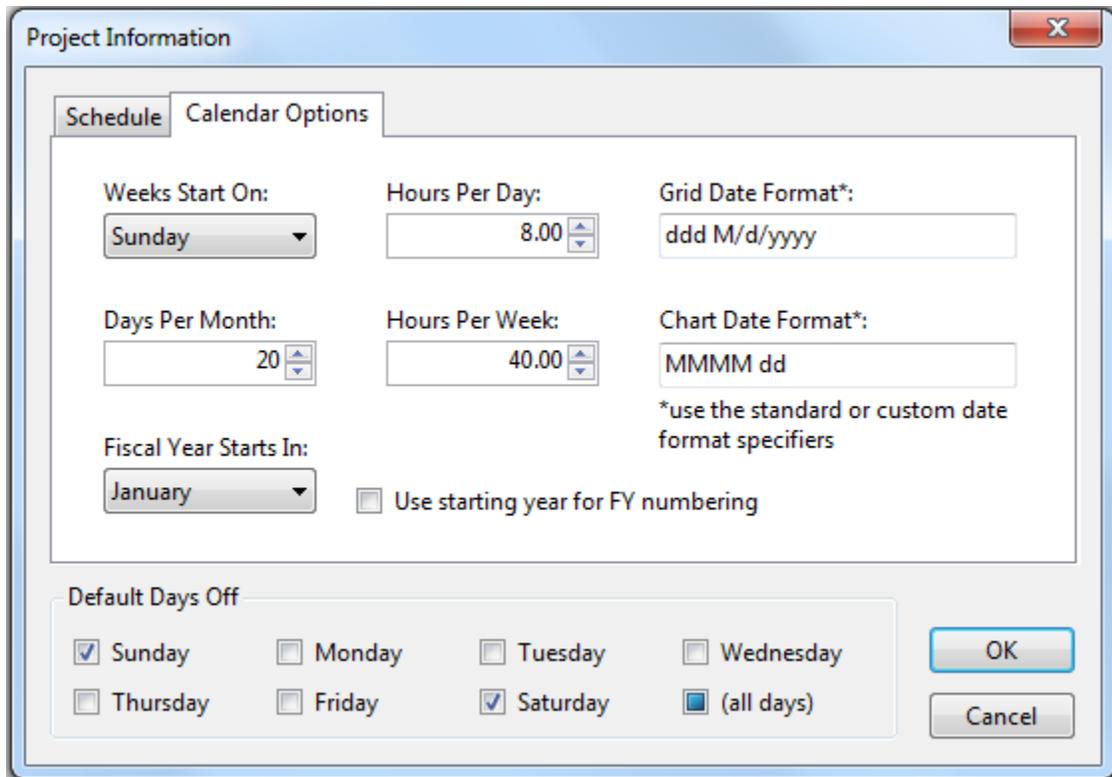
- **Schedule From:** dropdown listbox – Select either **Project Start Date** to schedule the project from the start date or **Project Finish Date** to schedule the project from the finish date. The default schedule begins with the project start date.

Note: If you need to find out how late you can start a project set the Schedule From to **Project Finish Date** and then once you begin the project set it back to **Project Start Date**.

- **Start Date:** dropdown listbox – Specify the start date if your Schedule From is specified to Project Start Date.
- **Finish Date:** dropdown listbox – Specify the finish date if your Finish Date: is specified to Project Finish Date.

In the **Default days off** groupbox select the days off from the project and in the **Default working time** groupbox specify the working hours.

When the **Calendar Options** tab is selected in the **Project Information** dialog box, the following items operate as follows:



The **Calendar Options** tab consists of the following items:

- **Weeks Start On:** – Indicates the day of the week that the scheduled task starts on. By default each week begins on Sunday.
- **Hours Per Day:** – Indicates the hours per day for the scheduled task. By default when C1GanttView calculates duration units, one day equals 8 hours.
- **Grid Date Format** – Specifies the format display for the grid in the C1GanttView control
- **Days Per Month** – Indicates the working days out of the normal calendar month for the automatically scheduled task. By default one month equals 20 working days.
- **Hours Per Week:** – Indicates the working hours per week of the automatically scheduled task. By default one working week is 40 hours.
- **Chart Date Format:** – Specifies the format display for the chart based on the standard or custom date format specifiers.
- **Fiscal Year Starts In:** – Specifies which month the fiscal year starts. The default month is January.
- **Use starting year for FY numbering** – Select this checkbox if you want to use the starting year for the Fiscal Year numbering. By default the next year is used for the fiscal year numbering. For example if the current fiscal year starts in april 2013 you can set the numbering of the fiscal year to 2014.

Project Resources Dialog Box

The **Project Resources dialog box** is used for adding or removing resources such as work, material, or cost to your project schedule.

Add a Resource

You can add a Resource by clicking the **Add** button. Once the Resource is added the default name, Resource1 appears in the Resource Name textbox. T

Remove a Resource

You can remove a resource by selecting the resource in the Members listbox and clicking on the **Remove** button.

Resource Name

This is where the name appears for your resource once it has been added. This is the name you'll refer to your resource in the project so it's advisable to type a descriptive and unique name for each resource.

Resource Type

In the Resource Type: dropdown list box you can specify three types of resources for your project schedule: work, material, and cost.

A work resource is anyone or anything that is needed to complete a project such as people and machines. Typically resources are people involved in your project whether or not they are assigned tasks. Equipment can include web servers or computers that have special software needs to accomplish certain tasks. Work resources need time (hours, days, weeks) to finish the task.

A material resource includes things that are consumed by a task such as paper, pens, and oil. They don't depend on the total work amount or duration of the task.

A cost resource is anything that doesn't depend on the total work amount or duration of the task such as dining and airfares. This type of resource is needed in your project to analyze your costs.

Unit of Measure:

The unit of measure for the resource type is specified in the **Unit of Measure:** textbox. Work resources are always measured by time so the typical unit of measure for a work resource is expressed in hours. The unit of measure for material resources is a measure unit like cubic yards, tons, meters, etc.

Notes

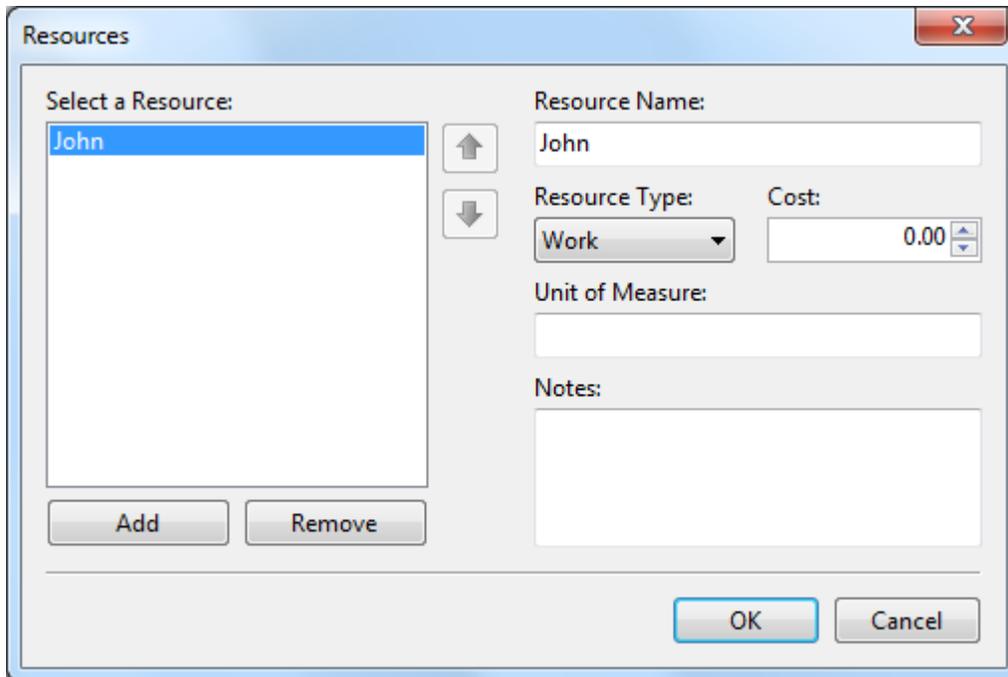
You can specify the language and concept details of the resource type in the **Notes:** textbox. This will help the user to better understand how the resource is calculated based upon the unit of measure.

Cost

You can specify the cost of the resource type using the numeric box.

To Access the Project Resources dialog box

Click on the **Project Resources** button , in the [C1GanttView Toolbar](#) (page 36).



Task Information Dialog Box

The **Task Information dialog box** is used for providing the details of the new task such as the task duration, task start date, task finish date, task schedule mode(manual or automatic), task percent completed, predecessor tasks, resource tasks, advanced tasks, and notes for the tasks. Styles for each task can also be modified by clicking on the **Styles** button.

To Access the Task Information dialog box

Click on the **Task information** button in the [C1GanttView Toolbar](#) (page 36).

See [Inserting a Task](#) (page 60) for an example.

Predecessors Tab

In the Predecessors tab you can add or remove predecessors, select the predecessor task name, select the predecessor type, and specify the lag time in days.

For conceptual information on predecessors see, [Task Predecessor](#) (page 49).

For procedural information on predecessors see, [Creating Predecessors](#) (page 64).

Task Information

Task Name:

Duration: 3.00 days

Schedule mode: Manually scheduled Auto scheduled

% Complete:

Start: 7/21/2012 12:00 AM

Finish: 7/26/2012 12:00 AM

Predecessors Resources Advanced Notes

Conception/Approval

Predecessor Task Name:

Predecessor Type:

Add Remove

Lag (in days):

Styles... OK Cancel

Resources Tab

In the Resources tab you can add a resource to the selected task using the **Add** button or remove a resource from the selected task using the **Remove** button.

Task Information

Task Name:

Duration: 14.00

Schedule mode: Manually scheduled Auto scheduled

% Complete:

Start: 7/16/2012 12:00 AM

Finish: 8/ 3/2012 12:00 AM

Predecessors Resources **Advanced** Notes

John

Resource Name:

Amount:

Cost: 0.00

Add Remove

Styles...

OK Cancel

Advanced Tab

In the **Advanced** tab you can specify the task calendar, whether or not to hide the selected task bar, and specify a deadline for the selected task. If the task is auto scheduled, you can choose the constraint type and constraint date for the selected task.

For more information on constraints see, [Task Constraints](#) (page 47).

Task Information

Task Name:

Duration: 4.00 days

Schedule mode: Manually scheduled Auto scheduled

% Complete:

Start: 7/16/2012 12:00 AM

Finish: 7/20/2012 12:00 AM

Predecessors Resources **Advanced** Notes

Task Calendar: (Default)

Constraint Type:

Constraint Date: Thu 7/19/2012

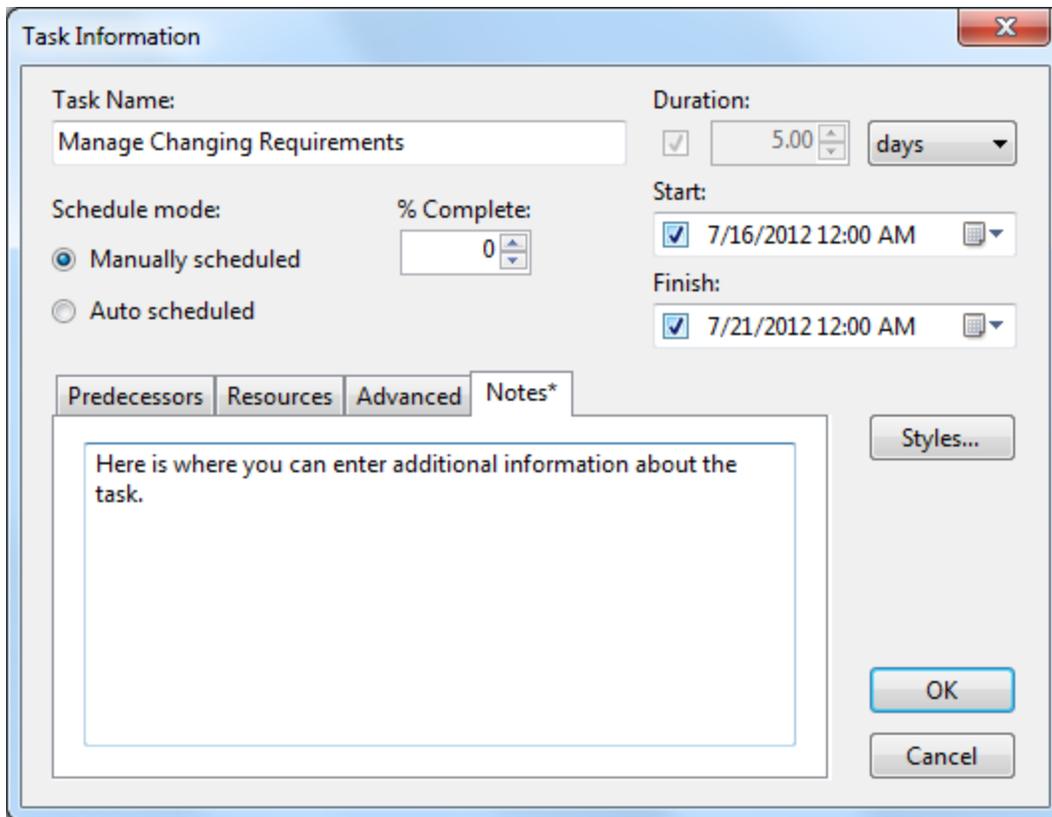
Hide task bar:

Deadline: Sun 7/15/2012

Styles... OK Cancel

Notes Tab

Additional comments about the task can be entered into the Notes textbox.

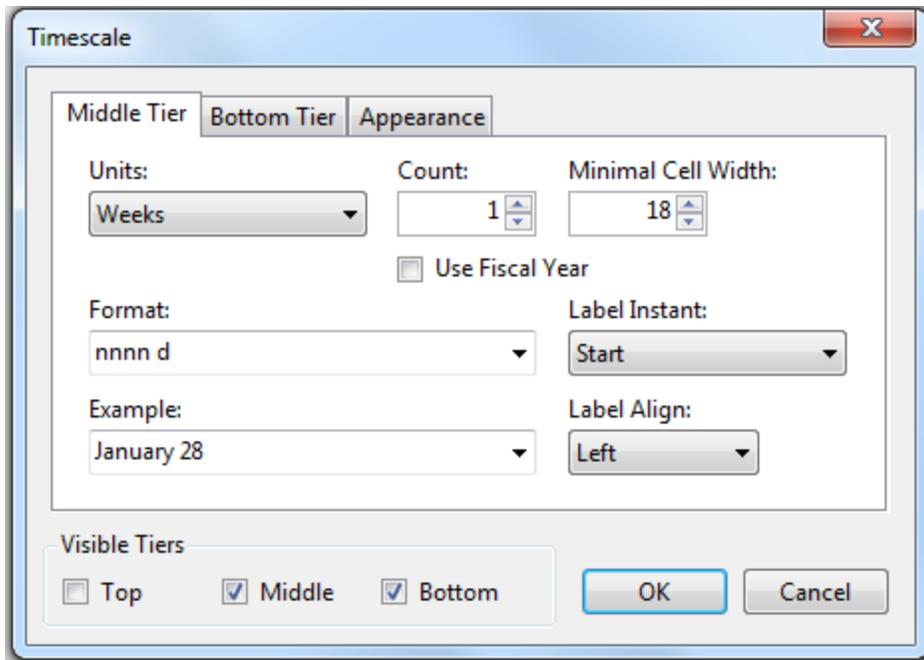


Timescale Dialog Box

The **Timescale** dialog box is used for setting the minimum date/time and maximum date/time, specifying visible tiers, and customizing the appearance of the current day, nonworking time, and/or the project start/finish time at run time.

To Access the Timescale dialog box

Click on the **Timescale** button in the [CIGanttView Toolbar](#) (page 36).



By default, two tiers are displayed, Middle and Bottom. When a tier (Top, Middle, or Bottom) is enabled a tab appears for each.

The items in the groupbox for each **Middle Tier**, **Bottom Tier**, or **Top Tier** tab consists of the following:

- The **Units**: dropdown listbox - specifies the time you wish to use.
- The **Count**: numeric box - Specifies a number to specify the frequency of unit labels on the timescale tier.
- The **Minimal Cell Width**: numeric box – Specifies the minimal cell width in pixels for each tier.
- The **Use Fiscal Year** checkbox- Specifies whether to base your timescale tier labels on the calendar year or fiscal year. Select the checkbox to base the timescale tier labels on your fiscal-year settings instead of your calendar year settings.
- The **Format**: dropdown listbox - Specifies the format you wish to use to display for the time unit. The default format is, nnnn d.
- The **Label Instant**: dropdown listbox – Specifies the where the instant label is placed: Start, End, Middle, Range, or Overlapped Range.
- The **Example**: dropdown listbox – Specifies the result of the date format that you entered in the Format: textbox above.
- The **Label Align**: dropdown listbox – Specifies the alignment of the specified tier, Left, Center, Right, or Justify.

See [Setting a Tier for the Timescale](#) (page 59) for an example.

C1GanttView Toolbar

The C1GanttView toolbar appears at the top of the **C1GanttView** control when the ShowToolbar property is set to **True**.



The CIGanttView toolbar consists of the following command buttons:

Command Button	Command Button Name	Description
	Load from xml file	Clicking on the Load from xml file button opens the Load from Xml File dialog box where you can browse to the location of the xml file you wish to load.
	Save as xml file	Clicking on the Save as xml file button opens the Save As Xml File dialog box where you can browse to the location of the xml file you wish to save.
	Grid Columns	Opens the Grid Columns dialog box (page 25).
	Move up task	Moves the selected task in the grid up one position.
	Move task down	Moves the selected task on position down.
	Task information	Opens the Task Information dialog box (page 30).
	Add task	Opens the Task Information dialog box (page 30) where the New Task is entered in the Task Name: textbox.
	Add blank row	Adds a blank row to task grid.
	Delete task	Deletes the selected task from the grid.
	Project information	Opens the Project Information dialog box (page 26).
	Change Working Time	Opens the Change Working Time dialog box (page 21).
	Progress Line	Clicking the Progress Line button opens the Progress Line dialog box (page 25).
	Project Resources	Opens the Project Resources dialog box (page 28).
	Timescale	Opens the Timescale dialog box (page 35).
	Bar styles	Opens the Bar Styles dialog box (page 20).
	Scroll to task	Scrolls to the task within the grid.

Timescale Formats

The following table details the available format specifiers for the timescale labels. For more information see the [Timescale Dialog Box](#) (page 35) and/or the Format property.

Format Specifier	Description
------------------	-------------

Standard Date/Time Formats	
s	s is the standard date/time format specifier. Ex: s(x) where 'x' is a placeholder for one of the following letters: d, D, f, g, m, t, y.
sd	Short date pattern (4/10/2008).
sD	Long date pattern (Thursday, April 10, 2008).
sf	Full date/time pattern (Thursday, April 10, 2008 6:30 AM).
sg	General date/time pattern (4/10/2008 6:30 AM).
sm	Month day pattern (April 10).
st	Short time pattern (6:30 AM).
sy	Year month pattern (April, 2008).
Year Formats	
yy	Represents the year as a two-digit number.
yyy	Represents the year with a minimum of three digits.
yyyy	Represents the year as a four-digit number.
Half Year Formats	
h	Represents a half-year as a number 1 or 2.
h{N1,N2}	Custom half-year name.
Quarterly Formats	
q	Represents a quarter as a number from 1 through 4.
q{N1,N2,N3,N4}	Custom quarter name.
Calendar Month Formats	
m	Represents the month as a number from 1 through 12.
mm	Represents the month as a number from 01 through 12.
n	Single-letter month name.
nnn	Use <code>DateTimeFormatInfo.GetAbbreviatedMonthName</code> .
nnnn	Use <code>DateTimeFormatInfo.GetMonthName</code> .
n{N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12}	Custom month name.
e{N1,N2,N3}	Custom thirds-of-month name.
Week of the Year Number Formats	
k	Represents the week of the year number from 1 to 53.

kk	Represents the week of the year number from 01 to 53.
Day of the Month Number Formats	
d	Represents the day of the month as a number from 1 through 31.
dd	Represents the day of the month as a number from 01 through 31.
Day of the Year Number Formats	
b	Represents the day of the year as a number from 1 through 366.
bbb	Represents the day of the year as a number from 001 through 366.
Single Letter Week Day Formats	
w	Single-letter week day (S, M, T, W,...)
ww	Use <code>DateTimeFormatInfo.GetShortestDayName</code> .
www	Use <code>DateTimeFormatInfo.GetAbbreviatedDayName</code> .
wwwwww	Use <code>DateTimeFormatInfo.GetDayName</code> .
w{N1,N2,N3,N4,N5,N6,N7}	Custom week day name.
Time Formats	
a	Represents the hour as a number from 1 through 12.
aa	Represents the hour as a number from 01 through 12.
u	Represents the hour as a number from 0 through 23.
uu	Represents the hour as a number from 00 through 23.
i	Represents the minute as a number from 0 through 59.
ii	Represents the minute as a number from 00 through 59.
t	Represents the first character of the AM/PM designator.
tt	Represents the AM/PM designator.
t{N1,N2}	Custom AM/PM designator.
Other Specifiers	
:	Separates the components of a time (use <code>DateTimeFormatInfo.TimeSeparator</code>)
/	Separates the components of a date (use <code>DateTimeFormatInfo.DateSeparator</code>)
“	Starts/ends a double quoted string (quotation

	mark).
'	Represents a quoted string (apostrophe).
\c	Displays the character 'c' as a literal.
Other character	Copies to the result string literally.

GanttView Appearance

C1GanttView is designed to make customization easy for you. Without writing any code, you can control the GanttView's appearance. The following topics provide information on the customizable elements within the C1GanttView.

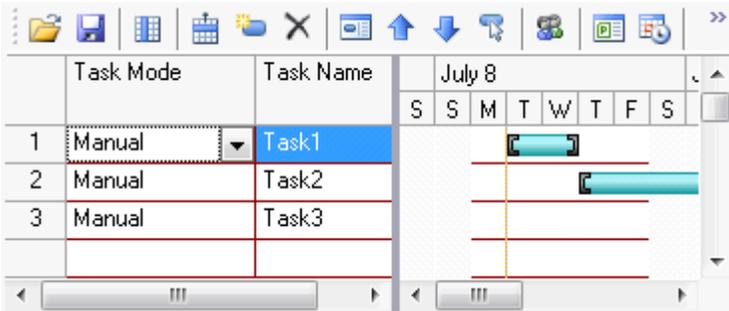
C1GanttView Appearance Properties

When the C1GanttView's **VisualStyle** property is set to **Custom** you can style any of its elements using the following properties:

Property	Description	Example
CellBorderColor	Specifies the custom color of cell borders in the grid and chart views.	CellBorderColor Example (page 41)
EmptyAreaBackColor	Specifies the background color of the empty area below the tasks.	
FixedBackColor	Specifies the custom background color of the fixed column/timescale area.	FixedBackColor Example (page 41)
FixedCellBorderColor	Specifies the custom color of cell borders in the fixed area.	FixedCellBorderColor Example (page 41)
FixedForeColor	Specifies the custom foreground color of the fixed column/timescale area.	FixedForeColor Example (page 41)
HighlightBackColor	Specifies the custom background color of the highlighted row in the grid view.	HighlightBackColor Example (page 42)
HighlightForeColor	Specifies the custom foreground color of the highlighted row in the grid view.	HighlightForeColor Example (page 42)
NonWorkingTimeColor	Specifies the color of the brush for drawing non-working time.	NonWorkingTimeColor Example (page 42)
SplitterColor	Specifies the custom color of the splitter between the grid and chart views	SplitterColor Example (page 43)
StartFinishLineColor	Specifies the color of the project start/finish date lines.	StartFinishLineColor Example (page 43)
TodayLineColor	Specifies the color of the "today" line.	TodayLineColor Example (page 43)
ToolbarBackColor	Specifies the custom background color of the toolbar.	ToolbarBackColor Example (page 44)

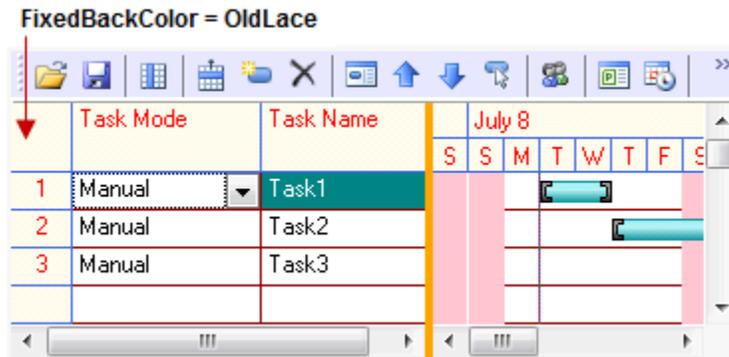
CellBorderColor Example

The following image illustrates the **CellBorderColor** property modified to DarkRed.



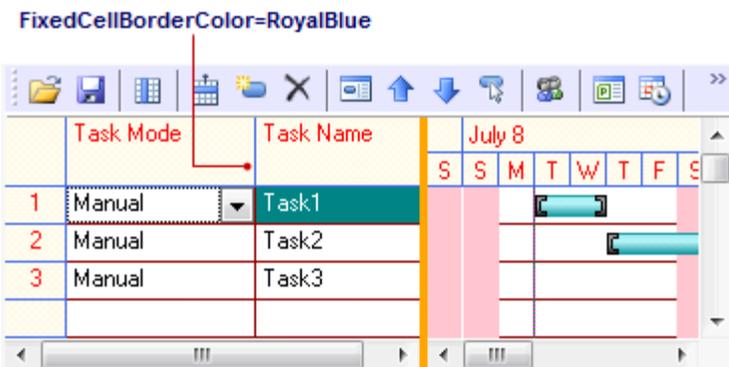
FixedBackColor Example

The following image illustrates the **FixedBackColor** property modified to OldLace.



FixedCellBorderColor Example

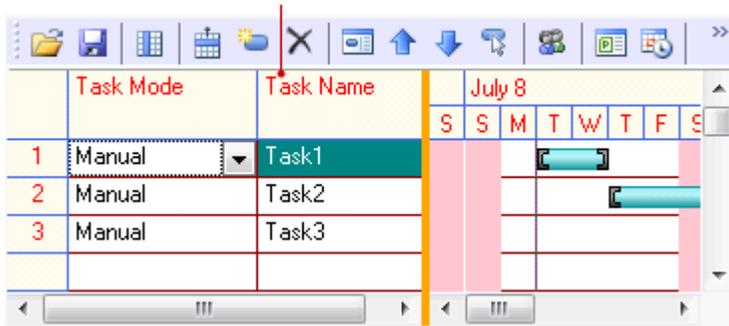
The following C1GanttView illustrates the **FixedCellBorderColor** modified to RoyalBlue.



FixedForeColor Example

The following C1GanttView illustrates the **FixedForeColor** modified to Red.

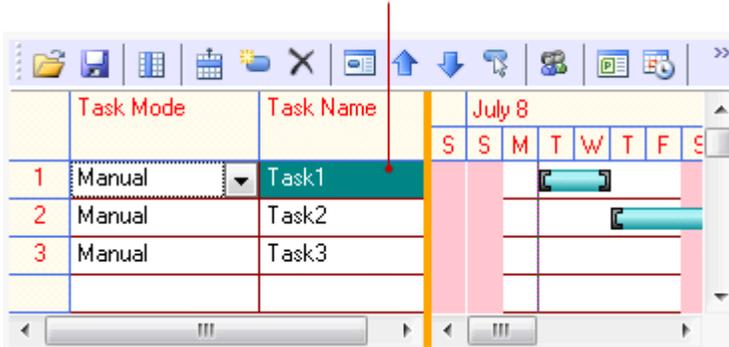
FixedForeColor=Red



HighlightBackColor Example

The following C1GanttView illustrates the **HighlightBackColor** modified to **Teal**.

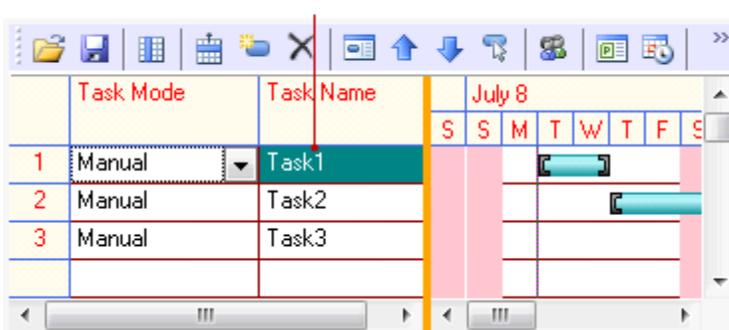
HighlightBackColor=Teal



HighlightForeColor Example

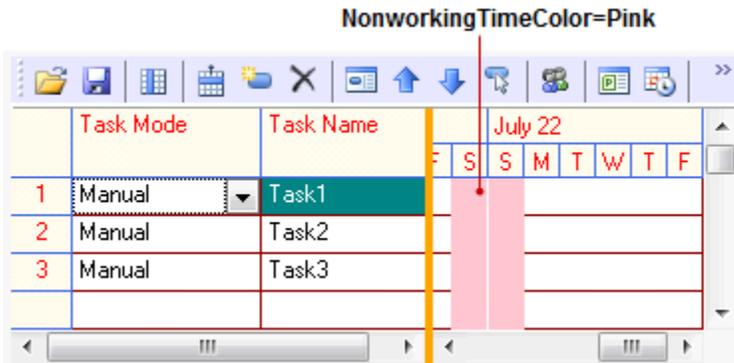
The following C1GanttView illustrates the **HighlightForeColor** modified to **HighlightText**.

HighlightForecolor=HighlightText



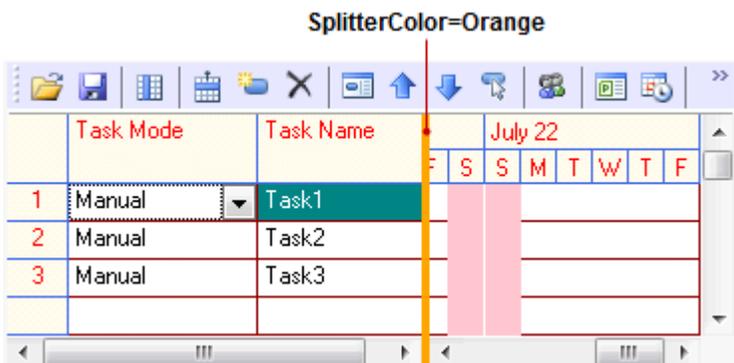
NonworkingTimeColor Example

The following C1GanttView illustrates the **NonworkingTimeColor** property modified to **Pink**.



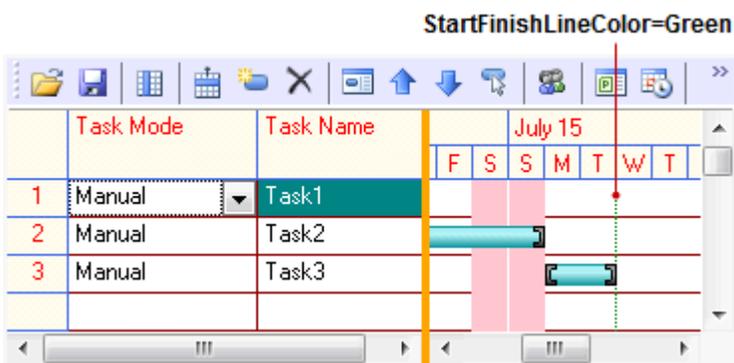
SplitterColor Example

The following C1GanttView illustrates the **SplitterColor** property modified to **Orange**.



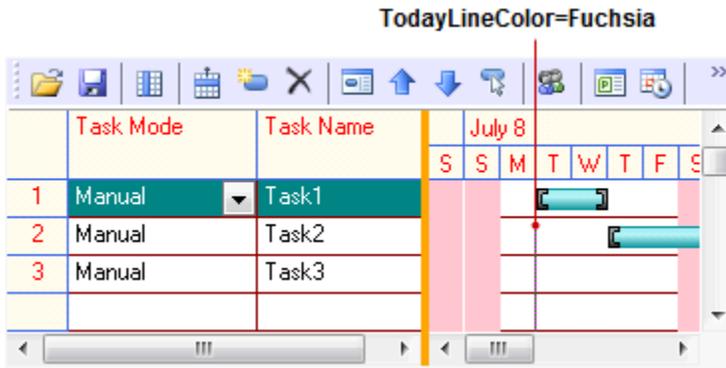
StartFinishLineColor Example

The following C1GanttView illustrates the **StartFinishLineColor** property modified to **Green**.



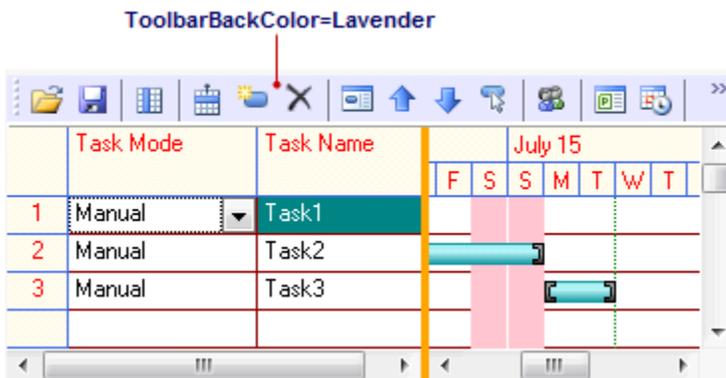
TodayLineColor Example

The following C1GanttView illustrates the **TodayLineColor** property modified to **Fuchsia**.



ToolbarBackColor Example

The following C1GanttView illustrates the **ToolbarBackColor** property modified to **Lavender**.



Optional Elements

You can determine whether or not to show the nonworking time, highlight start and finish dates, highlight today's date, and show the C1GanttView toolbar through the following properties:

- ShowNonworkingTime
- ShowStartFinish
- ShowToday
- ShowToolbar

Visual Styles

C1GanttView provides seven built-in styles with an additional custom style for the control – **Custom**, **Office2007Black**, **Office2007Blue**, **Office2007Silver**, **Office2010Black**, **Office2010Blue**, **Office2010Silver**, and **Windows7** – that can be easily applied to the control setting the **VisualStyle** property.

The following table illustrates each of the seven built-in visual styles:

Visual Style	Appearance
--------------	------------

Office2007Black

The screenshot shows the Office 2007 Black theme interface. At the top is a ribbon with icons for file operations and navigation. Below is a task list table with columns for Task Mode and Task Name. To the right is a Gantt chart for July 15, showing task bars for Task 1, Task 2, and Task 3. Task 1 is an auto task starting on Monday, Task 2 is an auto task starting on Tuesday, and Task 3 is a manual task starting on Wednesday.

	Task Mode	Task Name	July 15							
			F	S	S	M	T	W	T	F
1	Auto	Task 1				■	■			
2	Auto	Task 2					■	■		
3	Manual	Task 3						■	■	

Office2007Blue

The screenshot shows the Office 2007 Blue theme interface. It features a ribbon at the top and a task list table. The Gantt chart for July 15 shows Task 1 (Auto) starting on Monday, Task 2 (Auto) starting on Tuesday, and Task 3 (Manual) starting on Wednesday.

	Task Mode	Task Name	July 15							
			F	S	S	M	T	W	T	F
1	Auto	Task 1				■	■			
2	Auto	Task 2					■	■		
3	Manual	Task 3						■	■	

Office2007Silver

The screenshot shows the Office 2007 Silver theme interface. It includes a ribbon at the top and a task list table. The Gantt chart for July 15 displays Task 1 (Auto) starting on Monday, Task 2 (Auto) starting on Tuesday, and Task 3 (Manual) starting on Wednesday.

	Task Mode	Task Name	July 15							
			F	S	S	M	T	W	T	F
1	Auto	Task 1				■	■			
2	Auto	Task 2					■	■		
3	Manual	Task 3						■	■	

Office2010Black

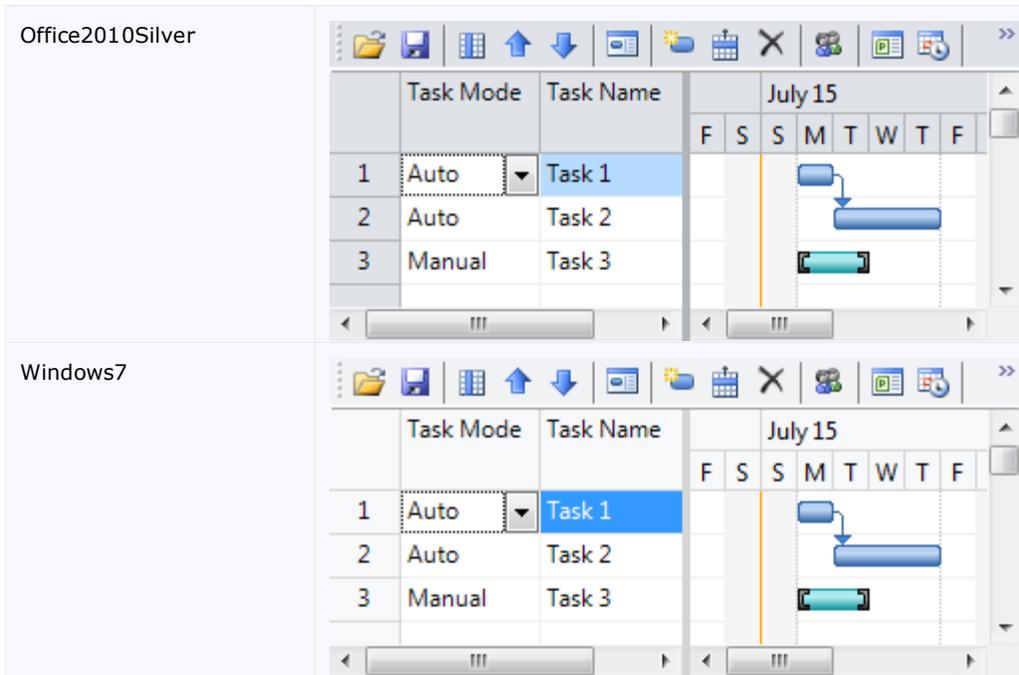
The screenshot shows the Office 2010 Black theme interface. It features a ribbon at the top and a task list table. The Gantt chart for July 15 shows Task 1 (Auto) starting on Monday, Task 2 (Auto) starting on Tuesday, and Task 3 (Manual) starting on Wednesday.

	Task Mode	Task Name	July 15							
			F	S	S	M	T	W	T	F
1	Auto	Task 1				■	■			
2	Auto	Task 2					■	■		
3	Manual	Task 3						■	■	

Office2010Blue

The screenshot shows the Office 2010 Blue theme interface. It includes a ribbon at the top and a task list table. The Gantt chart for July 15 displays Task 1 (Auto) starting on Monday, Task 2 (Auto) starting on Tuesday, and Task 3 (Manual) starting on Wednesday.

	Task Mode	Task Name	July 15							
			F	S	S	M	T	W	T	F
1	Auto	Task 1				■	■			
2	Auto	Task 2					■	■		
3	Manual	Task 3						■	■	



Bar Styles

By default a rectangular shaped bar represents each task. The shape, pattern, and color of the bar can be changed as well as the position of the bar text.

The bar type can be any of the following:

- Auto Scheduled
- Manually Scheduled
- Progress Bar
- Milestone
- Deadline
- Duration Only
- Start Only
- Finish Only

The Bar styles for each task can be modified through the Bar Styles dialog box. For more information see [Bar Styles Dialog Box](#) (page 20).

For more information see [Customizing the Bar Style](#) (page 73).

Task Elements

A task represents a fraction of the work that needs to be finished to complete the project. Each task can include some or all of the following constituents:

- [Task Duration](#) (page 48)
- [Task Mode](#) (page 47)

- [Task Deadline](#) (page 48)
- [Task Resources](#) (page 51)
- [Task Start and Finish Time](#) (page 50)
- [Task Constraints](#) (page 47)
- [Task Predecessor](#) (page 49)
- [Task Notes](#) (page 48)
- [Milestones](#) (page 52)

Task Mode

C1GanttView uses two types of methods to schedule tasks: manual scheduling and automatic scheduling. The following topics briefly describes each type of method.

Manual Tasks

The manual task is the default task type for C1GanttView. It provides greater flexibility for users planning and managing their schedule since it can be positioned anywhere in your schedule and the project won't move it. Manual tasks do not change the scheduling of the tasks for constraints, project resources, and dependencies. They are useful when project managers don't have complete information about each task. For example, they may know when the start date of the project is, but not its duration until they receive an estimation from their team members.

Automatic Tasks

Automatic scheduling provides a more structured way of managing project schedules. The GanttView automatically calculates the best earliest and latest dates for tasks once the user enters the information for the task duration, constraint dates, and number of resources. If something changes in the schedule such as the duration, task dependencies, or constraints, the GanttView project automatically adjusts the project schedule for you so you have the optimal schedule.

Changing Task Scheduling Modes

Each task can be changed back and forth from automatic to manual.

Changing a tasks' method can affect the project schedule in the following ways:

- Manual task changed to Automatic task will set the duration and start/finish dates to the GanttView project's default settings.
- Automatic task changed to Manual will keep its duration and start/finish dates.

Task Constraints

If your project is auto scheduled then you can add task constraints for each task using the ConstraintType and ConstraintDate properties. First you specify which type of constraint for the ConstraintType property. The ConstraintType property provides the following type of constraints:

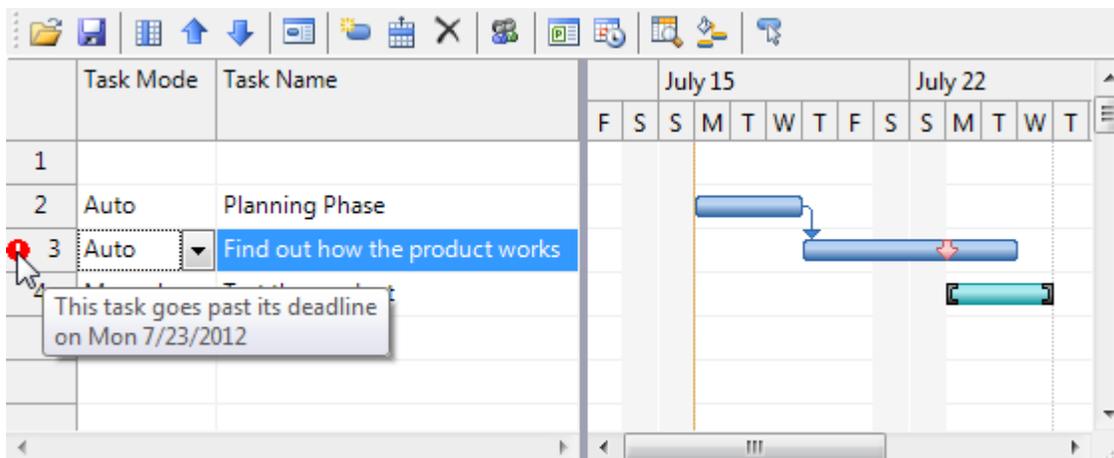
Constraint Type	Description
Default	No constraint.
StartNoEarlierThan	Specifies the earliest start date for the task. Use this constraint to make sure the task does not start before a specified date. This constraint is selected by default if the task is a successor.

StartNoLaterThan	Specifies the latest start date for the task. Use this constraint to make sure the task does not start after a specified date. This constraint is selected by default if the task is a predecessor.
FinishNoEarlierThan	Specifies the earliest finish date for the task. Use this constraint to make sure the task does not finish before a certain date.
FinishNoLaterThan	Specifies the latest finish date for the task. Use this constraint to make sure the task does not finish after a certain date.
MustStartOn	Specifies the date the task must begin.
MustFinishOn	Specifies the date the task must end.

Once you have selected a constraint type for the task you can specify the constraint date using the **ConstraintDate** property.

Task Deadline

You can specify the task deadline using the **Deadline** property. If your Finish date time exceeds your Deadline date time a red indicator will appear next to the task number like the following:



Task Duration

Each task has a duration. The duration specifies how long the task will take to complete. C1GanttView uses the **DurationUnits** property to specify the units for the task duration. The default duration unit is days. The units can be specified in Minutes, Hours, Days, Weeks, or Months. The **Duration** property is used to specify the whole integer value for the units, such as 2 for 2 weeks.

Task Notes

Can add notes to make use of further information for each task. Tasks can be entered in the **Notes** property. Within the richtextbox you can add some notes for the task. To make the task notes appear in the grid column at run time, click on the grid columns button in the toolbar and select **Notes** from the **Grid Columns** dialog box. For more information see [Adding a Note to the Task](#) (page 56).

Task Information

Task Name:

Duration: 5.00 days

Schedule mode: Manually scheduled Auto scheduled

% Complete:

Start: 7/16/2012 12:00 AM

Finish: 7/21/2012 12:00 AM

Predecessors Resources Advanced **Notes***

Here is where you can enter additional information about the task.

Styles...

OK

Cancel

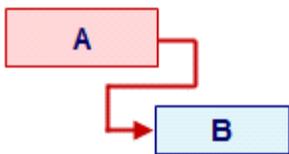
Task Predecessor

There are two types of task dependencies: predecessor and successor. A task predecessor's start or finish date determines the start or finish date of its successor task. A task successor's start or finish date is driven by its predecessor task. When you assign a predecessor task in C1GanttView the project automatically creates an arrow that points to the following task. If the predecessor type is not specified the **finish-to-start** dependency is created by default.

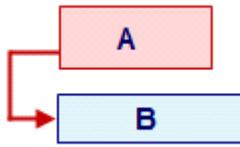
The task predecessor types can be specified in the **Task Information** dialog box. For information on how to create different predecessor types see, [Creating Predecessors](#) (page 64).

When there is an expected delay for something that must happen between two linked tasks, the lag time can be specified. For example, if you are waiting for a delivery of materials to arrive you can specify a lag time (in days). Days are the default unit for lag time.

The following table illustrates the different types of predecessors that can exist:

Task Image	Link Type	Description	Example
	Finish-to-start(FS)	This dependency is created by default when you link two tasks in C1GanttView. The order in which you select the tasks treats the first as the predecessor, the second as the successor. The work of task (B) can only start after all the work for task	For example, if you have two tasks, "Dig hole" and "Plant tree," the "Plant tree" task cannot begin until the "Dig hole" task is completed.

(A) is finished.



Start-to-start(SS)

The dependent task can start at any time after the task that it depends on begins. This dependency is used when two tasks can overlap or be done in parallel. When you overlap the tasks it will help reduce the total work time. For example if Task A will take 7 days and Task B will take 10 days then the overall time of the two tasks is only 10 days. The SS type does not require that both tasks begin at the same time.

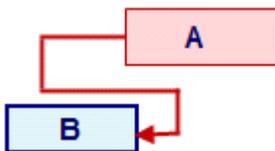
For example, if you have two tasks, “Planning phase” and “writing phase”, the “writing phase” task cannot begin until the “planning phase” begins.



Finish-to-finish (FF)

The finish date of task (A) determines the finish date of task (B). This dependency is used when two tasks can overlap or be done in parallel. When you overlap the tasks it will help reduce the total work time. For example if Task A will take 7 days and Task B will take 10 days then the overall time of the two tasks is only 10 days.

For example, if you have two tasks, “set up inner tent” and “snap together poles”, the “snap together poles” (Task B) cannot be completed until the “set up inner tent” (Task A) is completed.



Start-to-finish(SF)

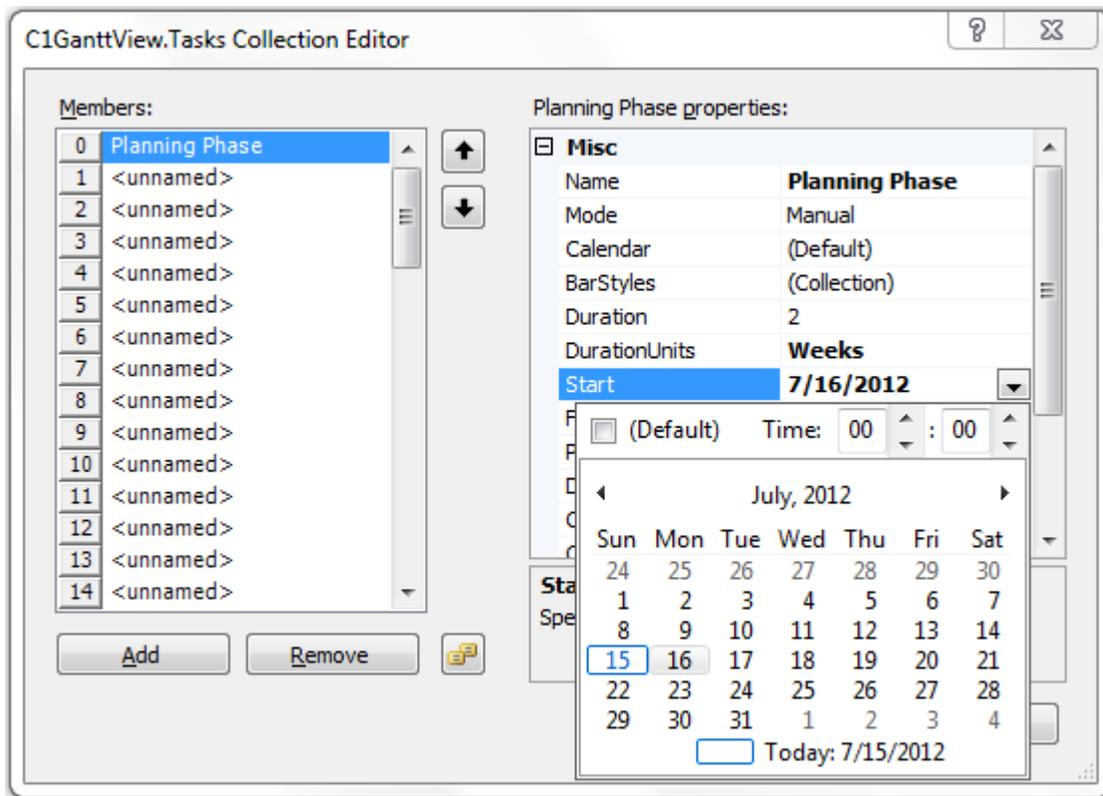
The start date of Task (A) determines the finish date of Task (B).

This type of dependency usually occurs less frequently.

For example, the bookshelves for your construction project are built off-site. Two of the tasks in your project are “Wood delivery” and “Assemble bookshelves”. The “assemble bookshelves” task cannot be completed until the “wood delivery” task begins.

Task Start and Finish Time

Generally each task has a start and finish time. All projects should have at least a start date. The Start time is specified by the **Start** property and can be expressed in **Date/Time** format. For your convenience a dropdown calendar appears when you click on the dropdown arrow next to the **Start** property at design-time or run-time where you can select the date from the dropdown calendar and set the time format in the increment button. The first increment button expresses the time in hours and the second increment button expresses the time in minutes. For example the start time for the Planning Phase task is set to 6/25/2012 at 8:00 a.m.



The Finish time is specified by the **Finish** property and can be expressed in DateTime format. For your convenience a dropdown calendar appears when you click on the dropdown arrow next to the **Finish** property at design-time or run-time where you can select the date from the dropdown calendar and set the time format in the increment button. The first increment button expresses the time in hours and the second increment button expresses the time in minutes.

You can schedule your project or task from the finish date if you need to do any of the following:

- Unsure when the project begins
- Would like to find out when the project should start by first entering a specific finish date
- Required to schedule the project from a finish date

Task Percent Complete

The **C1GanttView.PercentComplete** property specifies the percentage of the specified tasks that has been completed.

Task Resources

The following types of resources can be specified for your project schedule: work, material, and cost.

- Work resource - A work resource is anyone or anything that is needed to complete a project such as people and machines. Typically resources are people involved in your project whether or not they are assigned tasks. Equipment can include web servers or computers that have special software needs to accomplish certain tasks. Work resources need time (hours, days, weeks) to finish the task.
- Material resource - A material resource includes things that are consumed by a task such as paper, pens, and oil. They don't depend on the total work amount or duration of the task.

- Cost resource - A cost resource is anything that doesn't depend on the total work amount or duration of the task such as dining and airfares. This type of resource is needed in your project to analyze your costs.

For an example see [Assigning Resources to a Task](#) (page 57).

Milestones

A milestone is a significant point or landmark in your project. Milestones are created as tasks with zero duration and are visualized with a diamond shape. The milestone's shape and color are customizable. Additionally text can appear to the left, right, top, or bottom of the milestone marker.

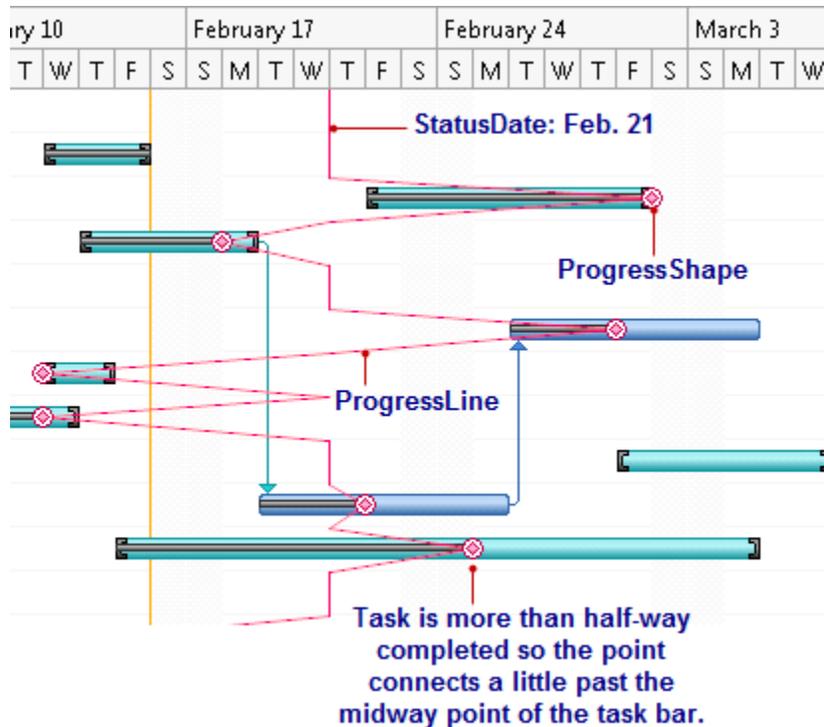
See [Creating a Milestone](#) (page 63) for more information.

Progress Lines

C1GanttView draws progress lines from the given fixed status date of your project that connect progress tasks and tasks that should have started. Progress lines are vertical lines on the timescale used to visually represent the progress of your project. The lines spike out to the left and right. The left spike indicates the work is behind schedule and the right spike indicates the work is ahead of schedule. The spike connects to the point on the taskbar that represents the task's percentage complete for its duration. In other words, if the task is 50% complete, the spike from the progress line connects to the middle of the taskbar. C1GanttView supports one progress line at a time. The progress line is drawn at the value of the StatusDate property.

Progress lines appear by default, but can be disabled by setting the Visible property to **False**. The progress lines color and style can be modified through the LineColor and LineStyle properties. The point's shape and color on the taskbar can be modified through the PointColor and PointShape properties.

The following image illustrates the progress lines:



To see how to modify the progress lines, see [Modifying the Progress Lines in your Project](#) (page 76).

GanttView for WinForms Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos which may make use of other development tools included with the ComponentOne Studios.

You can access samples from the **ComponentOne GanttView**. To view samples, click the **Start** button and then click **ComponentOne | Studio for WinForms | GanttView**. The following table provides a short overview of each sample.

Sample	Description
GanttViewDemo	Shows how to create a project timeline in C1GanttView by creating tasks, resources, assigning resources to specific tasks, and adding milestones to the project.

GanttView for WinForms Task-Based Help

The task-based help section assumes that you are familiar with programming in the Visual Studio NET environment and have a general understanding of the **ComponentOne GanttView** control.

Each topic provides a solution for specific tasks using the C1GanttView control. By following the steps outlined in each topic, you will be able to create projects using a variety of **C1GanttView** features.

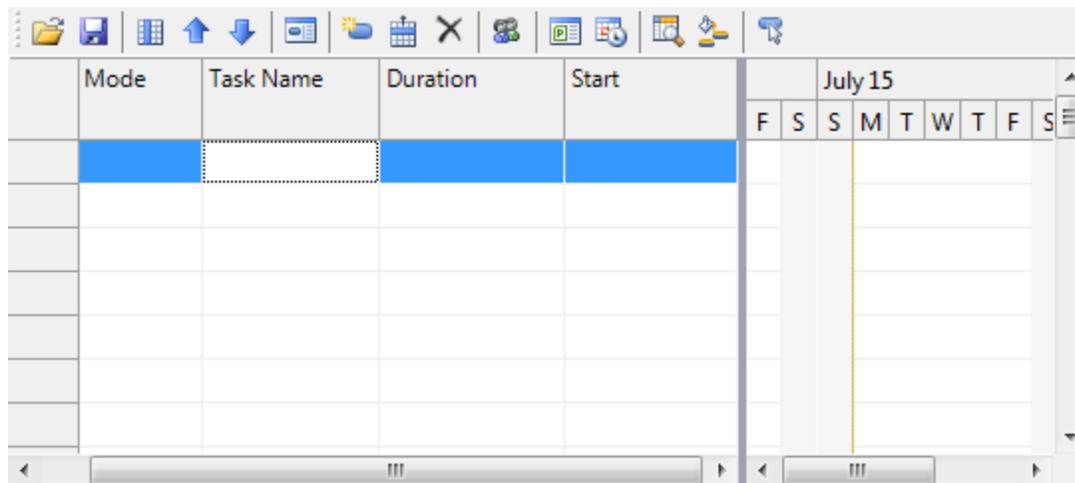
Adding Columns to the Grid

To add new columns to the grid at run time in the Gantt Chart complete the following:

1. Click the **Grid Columns** button, , to open the **Grid Columns** dialog box.
2. Select the checkbox next to **Duration** and **Start**.
3. Click **OK**.

This topic illustrates the following:

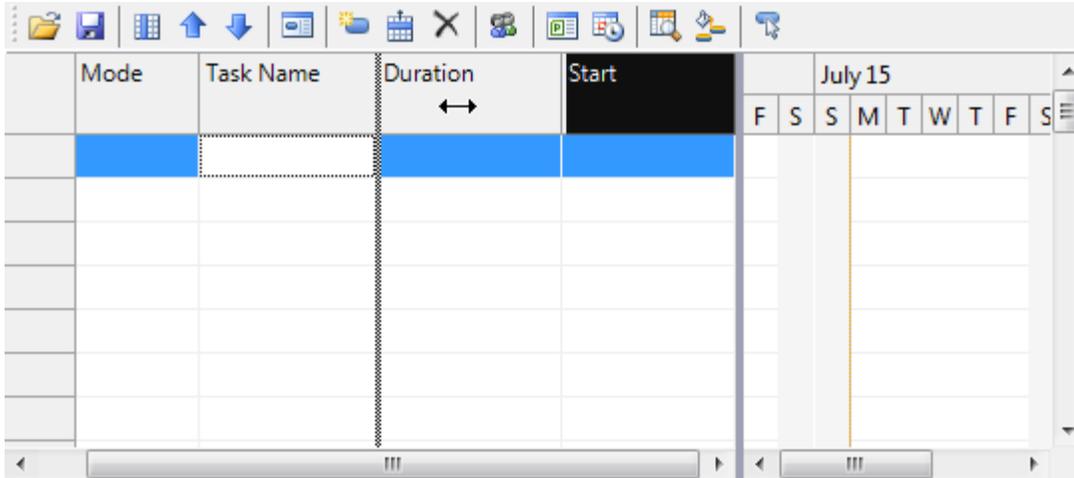
The following image shows a **C1GanttView** with the new grid columns, **Duration** and **Start**:



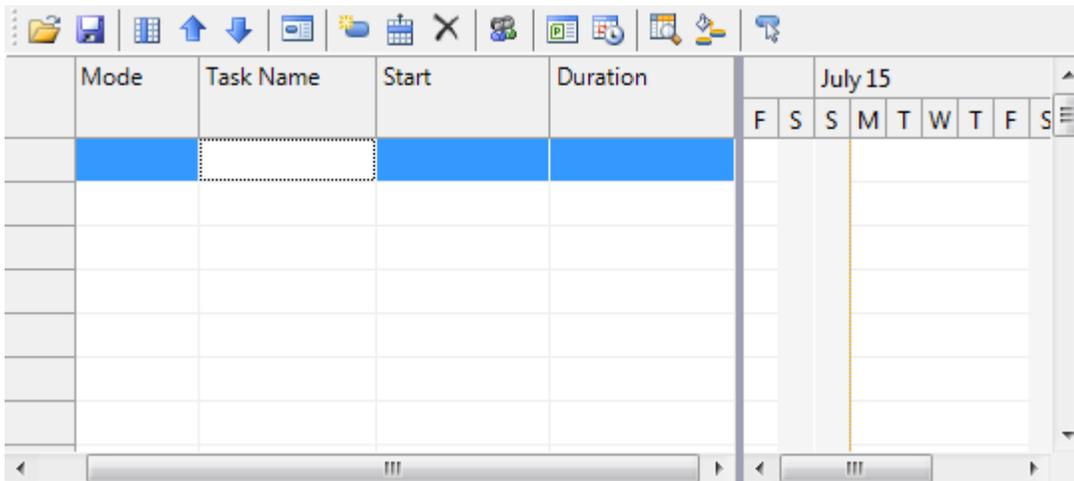
Moving Columns in the Grid

To move a column in the grid at run time complete the following:

1. Select the column you wish to move for example, **Start** column.



2. Drag the column and release the mouse button once you have positioned the column where you want it to be.

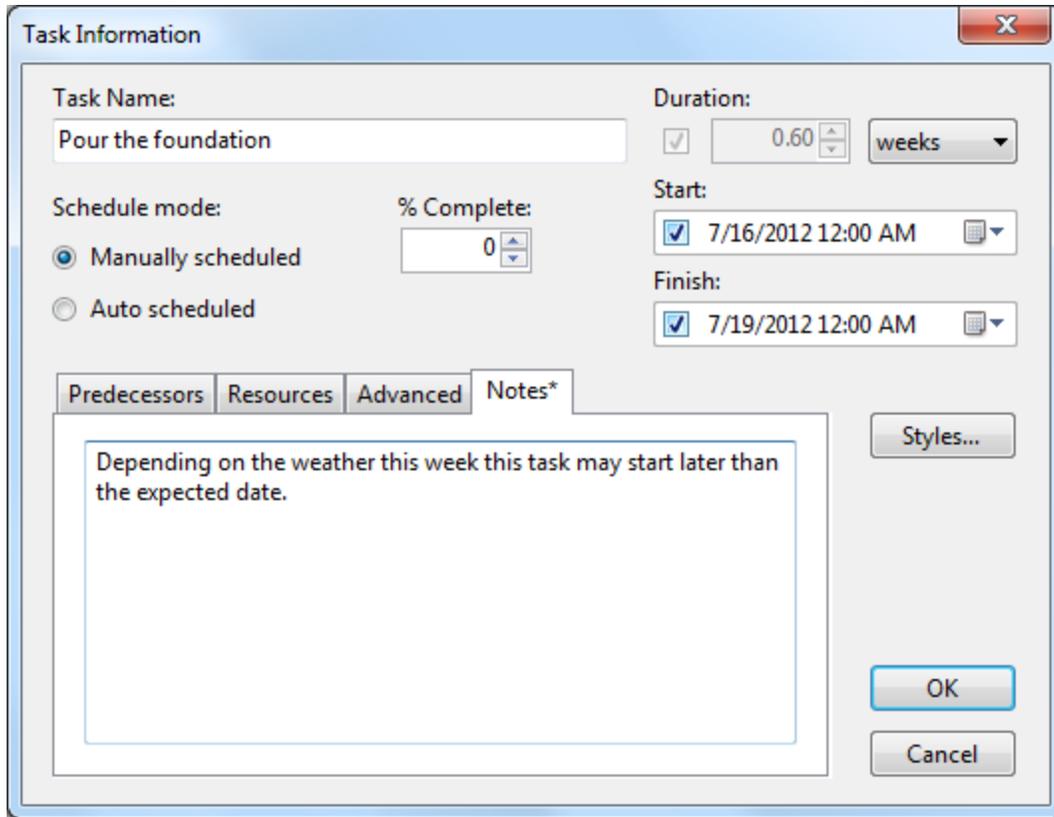


The **Start** column appears before the **Duration** column.

Adding a Note to the Task

This topic shows how to add a note to a specified task at run time.

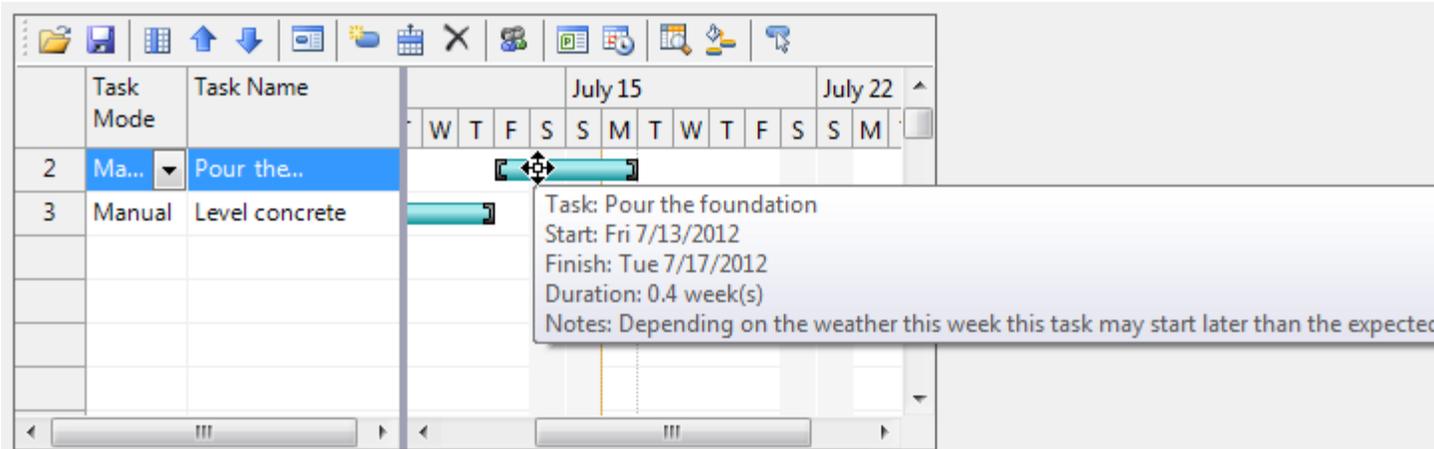
1. In the chart view, double click on the task bar you wish to add a note to.
2. In the **Task Information** dialog box, select the **Notes** tab and enter the information into the textbox.



3. Click **OK** to apply the changes to the selected task and to close the **Task Information** dialog box.

✔ **This topic illustrates the following**

When you hover over the task bar a tooltip appears with the information you entered into the **Notes** text box.



Assigning Resources to a Task

You can assign resources to a specific task first by creating the resource and then assigning the resource to the specific task. Resources can be created at design time, run time, or programmatically. At run time they can be

created using the **Project Resources** dialog box and then assigned to the task using the **Task Information** dialog box. At design time they can be created using the **C1GanttView.Resources Collection Editor** and then assigned to the task using the **Task.ResourceRefs Collection Editor**. Resources can also be created programmatically using the **Resource** class and **Add** method.

Add a resource to task1 at design time

To add a resource to task1 at design time, complete the following:

1. Click on the smart tag to open the **C1GanttView Tasks** menu.
2. Click on **Edit Resources** to open the **C1GanttView.Resources Collection Editor**.
3. The **C1GanttView.Resources Collection Editor** appears.
4. Click the **Add** button to add a resource to the collection.
5. Set **Resource 1 Name** to **Resource 1**.
6. Click **OK** to save and close the **C1GanttView.Resources Collection Editor**.
7. Right-click on the control and select **Edit Tasks**.
8. The **C1GanttView.Tasks Collection Editor** appears.
9. Select the **task1** task and click on the ellipsis button next to **ResourceRefs**.
10. The **Task.ResourceRefs Collection Editor** appears.
11. Click **Add** to add a reference to **Resource 1**.
12. Set the **Resource** to **Resource 1**.
13. Click **OK** to save and close the **Task.ResourceRefs Collection Editor**.

Add a resource to task1 programmatically

To programmatically add a resource to task1, complete the following:

- **C#**

```
private void btnAddResource_Click(object sender, EventArgs e)
{
    // add the new Resource object
    Resource r = new Resource();
    r.Name = "Resource 1";
    r.Cost = 300m;
    ganttView.Resources.Add(r);

    // find task1
    Task task1 = ganttView.Tasks.Search("Task 1");
    if (task1 != null && task1.ResourceRefs.Count == 0)
    {
        // add a resource reference to the task
        ResourceRef rRef = new ResourceRef();
        rRef.Resource = r;
        rRef.Amount = 0.5;
        task1.ResourceRefs.Add(rRef);
    }
}
```

- **Visual Basic**

```
Private Sub btnAddResource_Click(sender As Object, e As EventArgs)
    ' add the new Resource object
    Dim r As New Resource()
    r.Name = "Resource 1"
```

```

    r.Cost = 300D
    ganttView.Resources.Add(r)

    ' find task1
    Dim task1 As Task = ganttView.Tasks.Search("Task 1")
    If task1 IsNot Nothing AndAlso r IsNot Nothing AndAlso
task1.ResourceRefs.Count = 0 Then
        ' add a resource reference to the task
        Dim rRef As New ResourceRef()
        rRef.Resource = r
        rRef.Amount = 0.5
        task1.ResourceRefs.Add(rRef)
    End If
End Sub

```

Setting a Tier for the TimeScale

This topic shows you how to programmatically set the tier, specify the units, and format the timescale when you click on the **TimeScale** button.

- C#

```

private void btnTimescale_Click(object sender, EventArgs e)
{
    ScaleTier st = ganttView.Timescale.TopTier;
    st.Units = TimescaleUnits.ThirdsOfMonths;
    st.Format = "sd";
    st.Visible = true;
}

```

- Visual Basic

```

Private Sub btnTimescale_Click(sender As Object, e As EventArgs)
    Dim st As ScaleTier = ganttView.Timescale.TopTier
    st.Units = TimescaleUnits.ThirdsOfMonths
    st.Format = "sd"
    st.Visible = True
End Sub

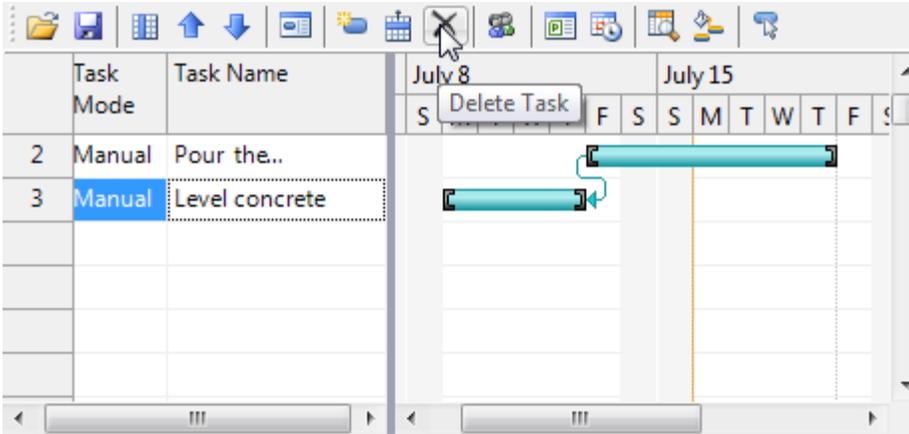
```

Deleting a Task

This topic shows how to delete a task at run time or in code. At run time, you can delete a task by clicking on the **Delete** button from the **C1GanttView** toolbar or you can use the index to programmatically specify the position of the new task.

Delete a task at run time

1. In the grid select the task you wish to delete.
2. Click on the **Delete** button in the C1GanttView toolbar.



The selected task is removed from the C1GanttView.

Delete a task programmatically

To programmatically delete a task, complete the following:

- C#

```
private void btnDelete_Click(object sender, EventArgs e)
{
    TaskCollection tasks = ganttView.Tasks;

    // find NewTask
    int index = tasks.IndexOf("New Task");
    if (index >= 0)
    {
        // delete and dispose the new task
        Task t = tasks[index];
        tasks.RemoveAt(index);
        t.Dispose();
    }
}
```

- Visual Basic

```
Private Sub btnDelete_Click(sender As Object, e As EventArgs)
    Dim tasks As TaskCollection = ganttView.Tasks

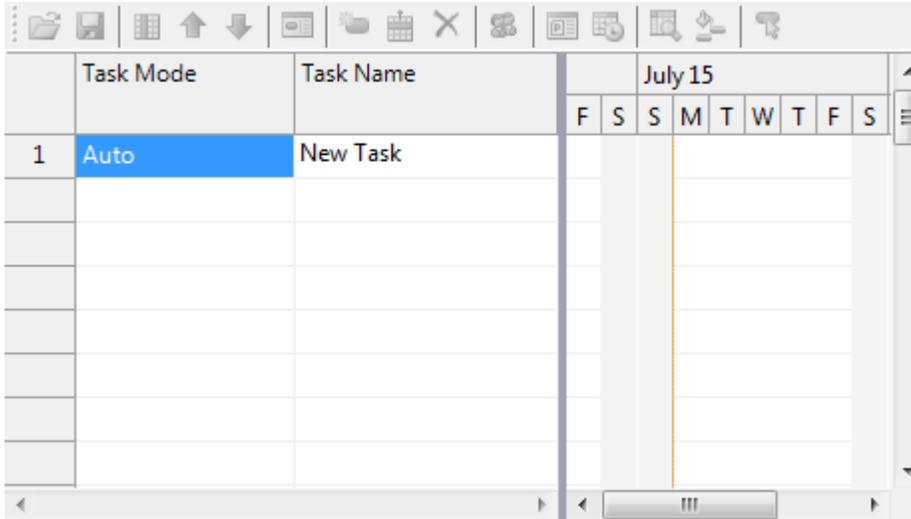
    ' find NewTask
    Dim index As Integer = tasks.IndexOf("New Task")
    If index >= 0 Then
        ' delete and dispose the new task
        Dim t As Task = tasks(index)
        tasks.RemoveAt(index)
        t.Dispose()
    End If
End Sub
```

Inserting a Task

This topic illustrates how to insert a task at run time or in code. At run time, you can insert a task between existing tasks by selecting the row below where you want a new task to appear or you can use the index to programmatically specify the position of the new task.

Insert a task at run time

1. In the **Task Name** field of the grid, type a task name at the end of the task list.
Notice as you type the C1GanttView toolbar items will become disabled.



2. Press **ENTER** and the New Task item will appear in the grid. The C1GanttView toolbar items will become enabled.

Insert a task in code

To programmatically insert a task, complete the following:

- C#

```
private void btnInsertTask_Click(object sender, EventArgs e)
{
    TaskCollection tasks = ganttView.Tasks;
    int index = tasks.IndexOf("Task 2");
    if (index >= 0)
    {
        // create a new task
        Task t = new Task();
        tasks.Insert(index, t);
        t.Mode = TaskMode.Automatic;
        t.Name = "New Task";
        t.Start = new DateTime(2012, 6, 25);
        t.Duration = 3;
    }
}
```

- Visual Basic

```
Private Sub btnInsertTask_Click(sender As Object, e As EventArgs)
    Dim tasks As TaskCollection = ganttView.Tasks
    Dim index As Integer = tasks.IndexOf("Task 2")
    If index >= 0 Then
        ' create a new task
        Dim t As New Task()
        tasks.Insert(index, t)
        t.Mode = TaskMode.Automatic
    End If
End Sub
```

```

t.Name = "New Task"
t.Start = New DateTime(2012, 6, 25)
t.Duration = 3
End If
End Sub

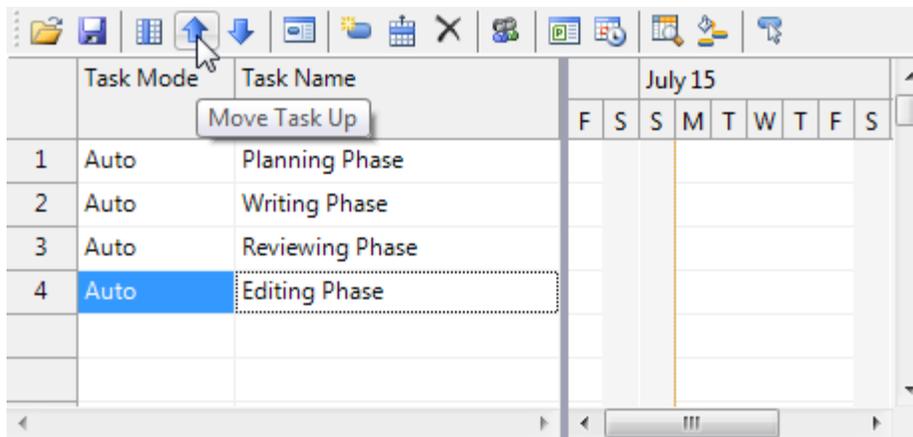
```

Moving a Task

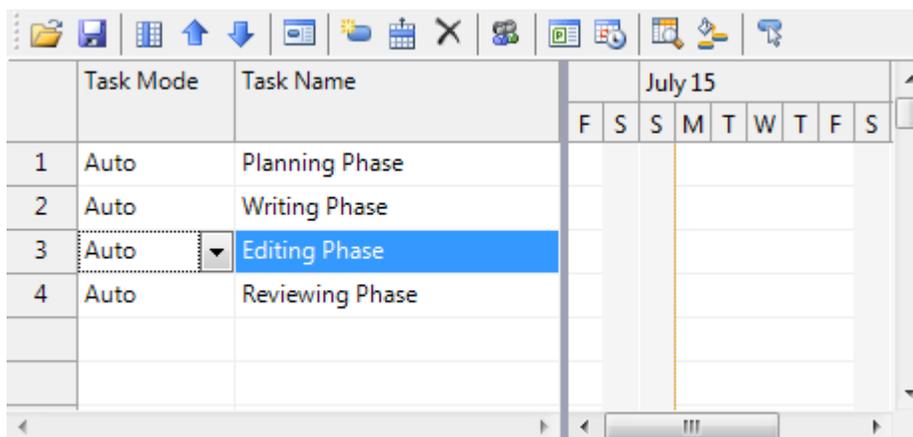
You can move a task at run time using the up or down arrows located on the CIGanttView toolbar or you can move tasks programmatically using the **RemoveAt** and **Insert** methods.

Move a task at run time

1. In the grid, select the task you wish to move.
2. Click either the **Move Task Up** button to move the task up or **Move Task Down** button to move the task down a position. In this example, we'll click the **Move Task Up** once to move the task item up once.



After clicking the **Move Task Up** button the **Editing phase** task is moved up one position.



Move a task programmatically

To move a task programmatically, complete the following:

- **C#**

```
private void btnMove_Click(object sender, EventArgs e)
{
    TaskCollection tasks = ganttView.Tasks;
    int index = tasks.IndexOf("New Task");
    if (index > 0)
    {
        Task t = tasks[index];
        tasks.RemoveAt(index);
        tasks.Insert(0, index - 1);
    }
}
```

- **Visual Basic**

```
Private Sub btnMove_Click(sender As Object, e As EventArgs)
    Dim tasks As TaskCollection = ganttView.Tasks
    Dim index As Integer = tasks.IndexOf("New Task")
    If index > 0 Then
        Dim t As Task = tasks(index)
        tasks.RemoveAt(index)
        tasks.Insert(0, index - 1)
    End If
End Sub
```

Creating a Milestone

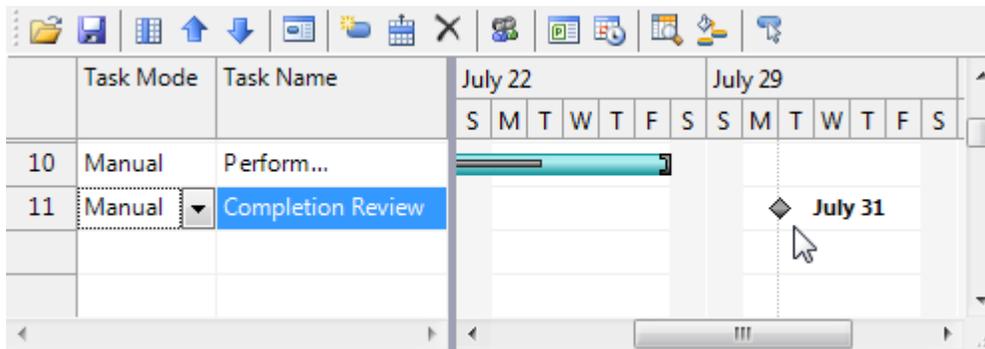
This topic shows how to create a milestone at design time.

To create a milestone at design time, complete the following:

1. Select the **C1GanttView** control and click on its smart tag to open the **C1GanttView Tasks** menu.
2. Select **Edit Tasks**.
The **C1GanttView.Tasks Collection** editor appears.
3. Click **Add** to add a task to the Members list.
4. Enter a Name next to the **Name** textbox, for example, **Completion Review**.
5. In the **Start** textbox enter the finish date of the project.
6. In the **Finish** textbox enter the finish date of the project.
7. Set the **Duration** to 0.

This topic illustrates the following:

A milestone with a diamond shape and the finish date text to the right of it appears in the chartview area of the ganttview.



Creating Predecessors

This topic shows how to create different types of predecessor tasks at run time through the **Task Information** dialog box or programmatically through the **Add** method.

Adding a Predecessor Programatically

To programmatically add a predecessor, complete the following:

- C#

```
private void btnAddPredecessor_Click(object sender, EventArgs e)
{
    // find task1 and task2
    Task task1 = ganttView.Tasks.Search("Task 1");
    Task task2 = ganttView.Tasks.Search("Task 2");

    if (task1 != null && task2 != null && task2.Predecessors.Count == 0)
    {
        // switch to auto-scheduling mode
        task2.Mode = TaskMode.Automatic;

        Predecessor p = new Predecessor();
        p.PredecessorTask = task1;
        task2.Predecessors.Add(p);

        // restore the manual mode
        task2.Mode = TaskMode.Manual;
    }
}
```

- Visual Basic

```
Private Sub btnAddPredecessor_Click(sender As Object, e As EventArgs)
    ' find task1 and task2
    Dim task1 As Task = ganttView.Tasks.Search("Task 1")
    Dim task2 As Task = ganttView.Tasks.Search("Task 2")

    If task1 IsNot Nothing AndAlso task2 IsNot Nothing AndAlso
    task2.Predecessors.Count = 0 Then
        ' switch to auto-scheduling mode
        task2.Mode = TaskMode.Automatic

        Dim p As New Predecessor()
        p.PredecessorTask = task1
        task2.Predecessors.Add(p)
    End If
End Sub
```

```

        ' restore the manual mode
        task2.Mode = TaskMode.Manual
    End If
End Sub

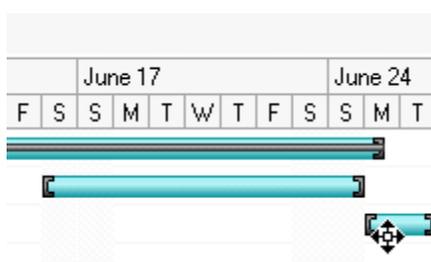
```

Creating a Finish to Start Predecessor Type

To create a finish to start predecessor type, use the **Task Information** dialog box at run time like the following:

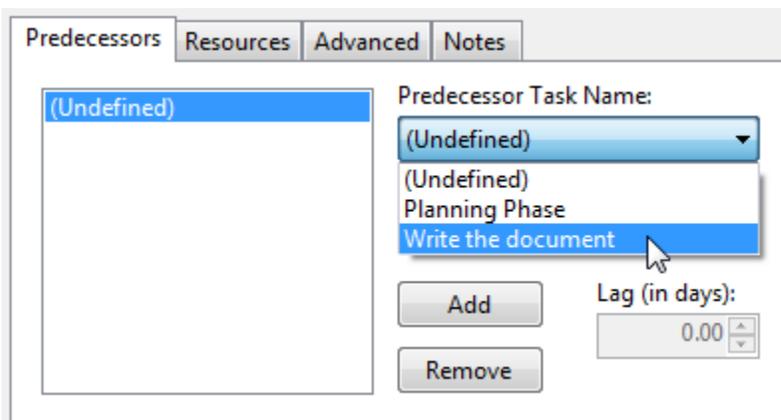
1. Double click on the task bar you wish to create the predecessor type in the chart view area, for example double-click on the **Edit the document** task.

The cursor icon will change once you place it over the desired task bar.



The [Task Information](#) (page 30) dialog box appears once you double-click the desired task.

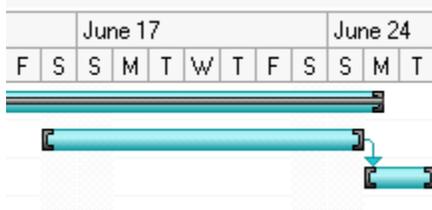
2. Select the **Predecessors** tab and click the **Add** button.
3. Click the dropdown arrow from the **Predecessor Task Name**: and select the predecessor task, for example, **Write the document**.



4. Select **Finish-to-Start (FS)** from the **Predecessor Type**.
5. Click **OK** to apply the changes and close the **Task Information** dialog box.

This topic illustrates the following:

A downward arrow pointing to the successor **Edit the document** of task **Write the document**.

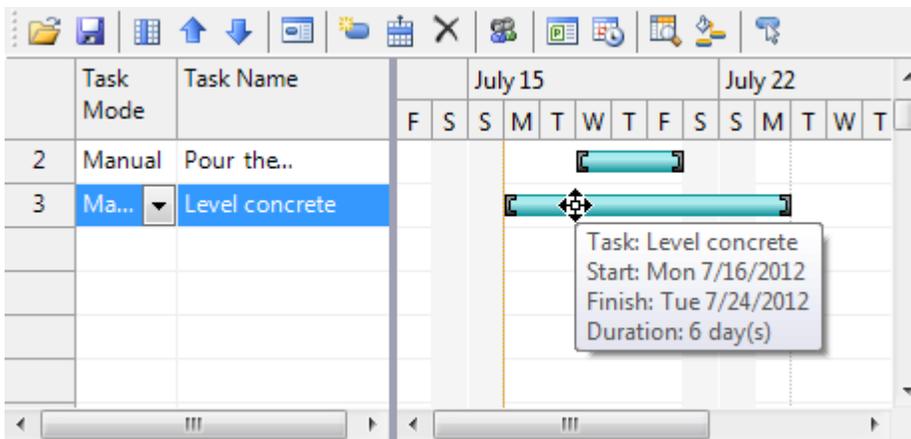


Creating a Start to Start Predecessor Type

To create a start-to-start predecessor type, use the **Task Information** dialog box at run time like the following:

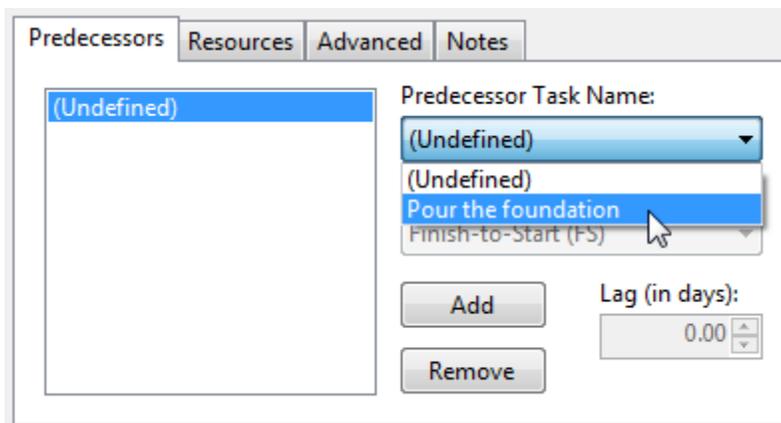
1. Double click on the task bar you wish to create the predecessor type in the chart view area, for example double-click on the **Level the concrete** task.

The cursor icon will change once you place it over the desired task bar.



The [Task Information](#) (page 30) dialog box appears once you double-click the desired task.

2. Select the **Predecessors** tab and click the **Add** button.
3. Click the drop-down arrow from the **Predecessor Task Name:** and select the predecessor task, for example, **Pour the foundation**.



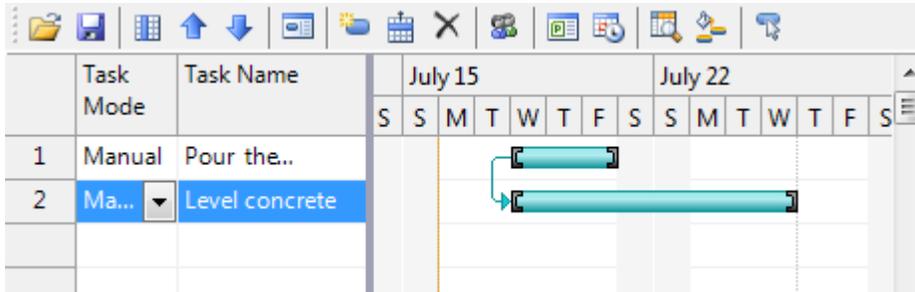
4. Select **Start-to-Start (FS)** from the **Predecessor Type**.

This predecessor type implies that the dependent task, **Level the concrete**, can start any time after the task, **Pour the foundation** that it depends on begins.

5. Click **OK** to apply the changes and close the **Task Information** dialog box.

✔ **This topic illustrates the following:**

The dependent task, **Level concrete**, can start at any time after the, **Pour the foundation**, task that it depends on begins. A downward arrow is drawn to illustrate the link between the two tasks.

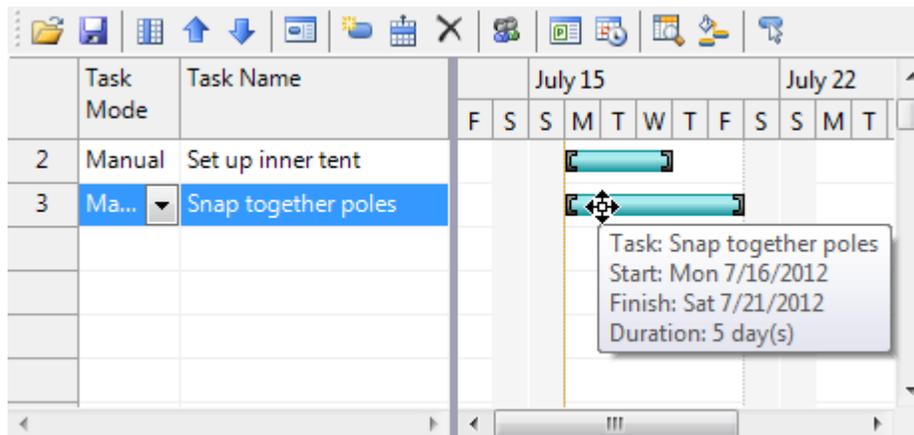


Creating a Finish to Finish Predecessor Type

To create a **finish-to-finish** predecessor type, use the **Task Information** dialog box at run time like the following:

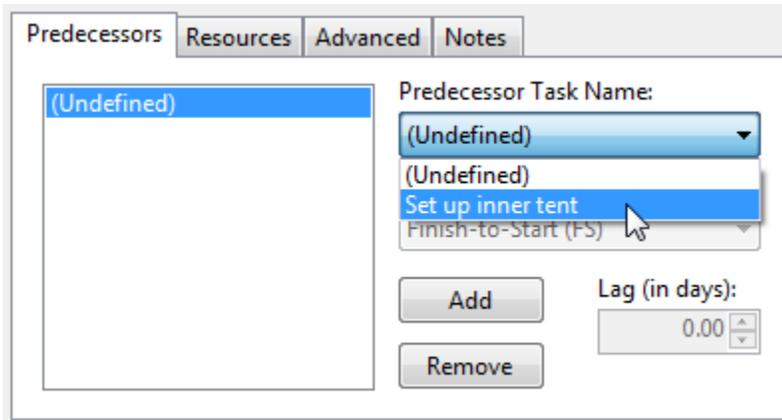
1. Double click on the task bar you wish to create the predecessor type in the chart view area, for example, **Snap together poles** task.

The cursor icon will change once you place it over the desired task bar.



The [Task Information](#) (page 30) dialog box appears once you double-click the desired task.

2. Select the **Predecessors** tab and click the **Add** button.
3. Click the dropdown arrow from the **Predecessor Task Name**: and select the predecessor task, for example, **Set up inner tent**.



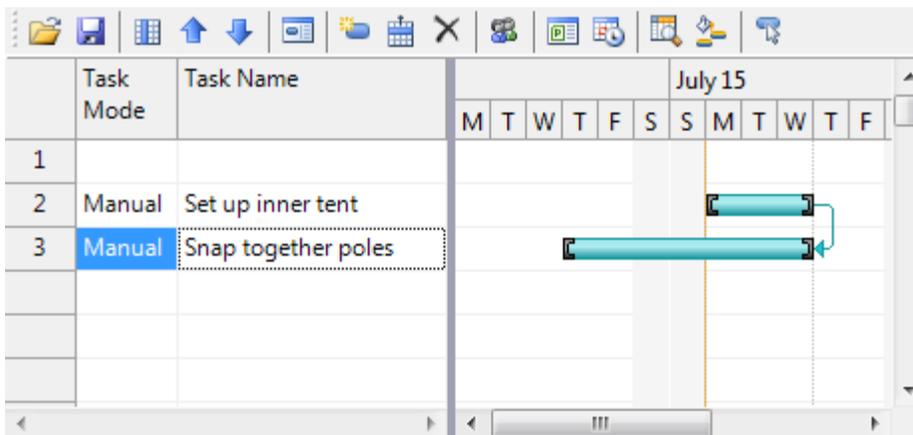
4. Select **Finish-to-Finish (FS)** from the **Predecessor Type**.

This predecessor type implies that the dependent task, **Snap together poles**, can't finish until the **Set up inner tent**, task finishes.

5. Click **OK** to apply the changes and close the **Task Information** dialog box.

✔ **This topic illustrates the following:**

The dependent task, **Snap together poles**, can start at any time after the **Set up inner tent** task that it depends on begins. A downward arrow is drawn to illustrate the link between the two tasks.

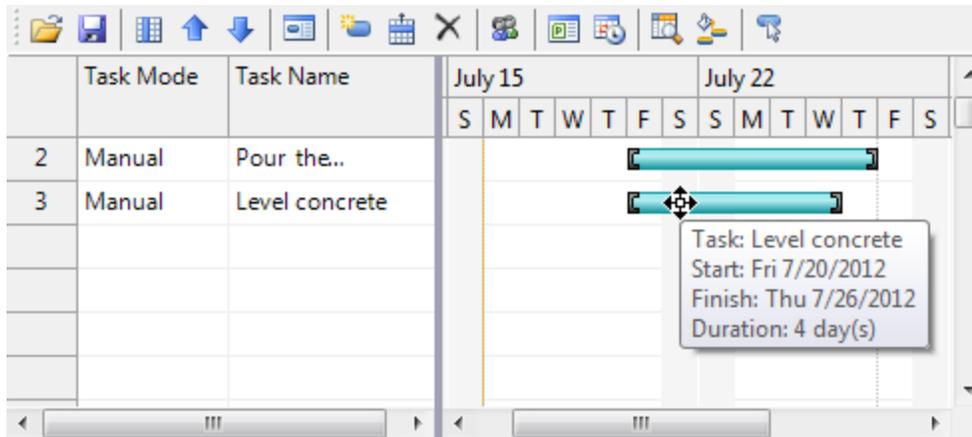


Creating a Start to Finish Predecessor Type

To create a **Start-to-Finish** predecessor type, use the **Task Information** dialog box at run time like the following:

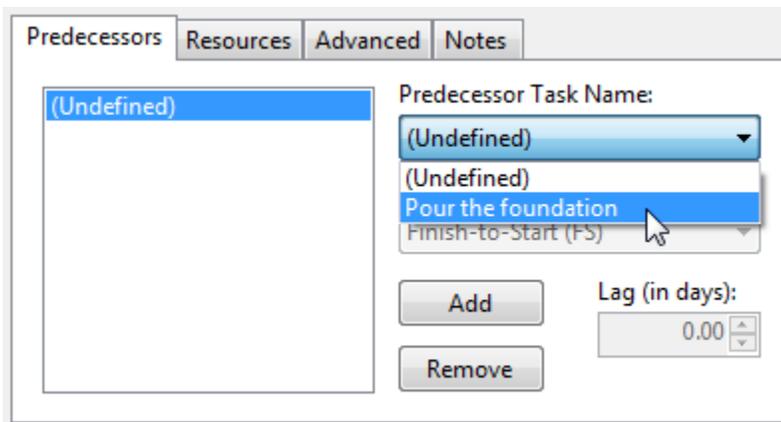
1. Double click on the task bar you wish to create the predecessor type in the chart view area, for example, **Level the concrete**, task.

The cursor icon will change once you place it over the desired task bar.



The [Task Information](#) (page 30) dialog box appears once you double-click the desired task.

2. Select the **Predecessors** tab and click the **Add** button.
3. Click the dropdown arrow from the **Predecessor Task Name:** and select the predecessor task, for example, **Pour the foundation**.



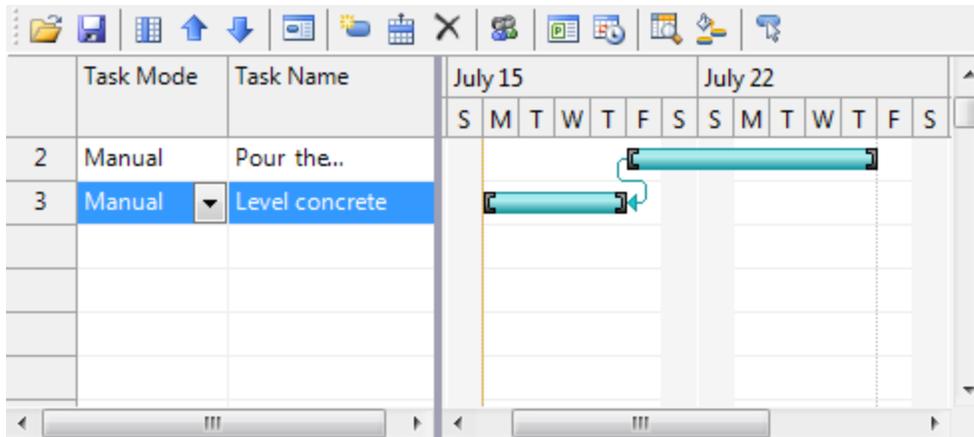
4. Select **Start-to-finish (SF)** from the **Predecessor Type**.

This predecessor type implies that the dependent task, **Level the concrete**, cannot finish until the start of, **Pour the foundation**.

5. Click **OK** to apply the changes and close the **Task Information** dialog box.

✔ **This topic illustrates the following:**

The dependent task, **Level concrete**, cannot finish until the start of **Pour the foundation**. A downward arrow is drawn to illustrate the link between the two tasks.



Adding a Vacation Day for a Resource

This topic illustrates how to add a calendar exception in Design view.

1. Click **C1GanttView**'s smart tag to open the **C1GanttView Tasks** menu and click **Edit Calendars**.
The **C1GanttView.CustomCalendars** Collection Editor appears.
2. Click **Add** to add a new calendar member to the editor.
3. Click on the ellipsis button next to the **CalendarExceptions** to open the **CustomCalendar.CalendarExceptions** Collection Editor.
4. Click **Add** to add a new calendar exception to the collection editor.
5. In the properties pane, enter, **My Holidays**, next to the **Name** property.
6. Expand the **RecurrencePattern** node and click on the dropdown arrow next to **WeekDays**.
7. Select the Monday checkbox.
8. Set the **WeekOfMonth** to Fourth.
9. Set the **StartDate** to 6/4/2012.
10. Click **OK** to save and close the **CustomCalendar.CalendarExceptions** Collection Editor.
11. Click **OK** to save and close the **C1GanttView.CustomCalendars** Collection Editor.

Saving and Loading GanttView as an XML File

This topic shows how to save the **C1GanttView** project as an XML file and how to load an existing **C1GanttView** project from an xml file.

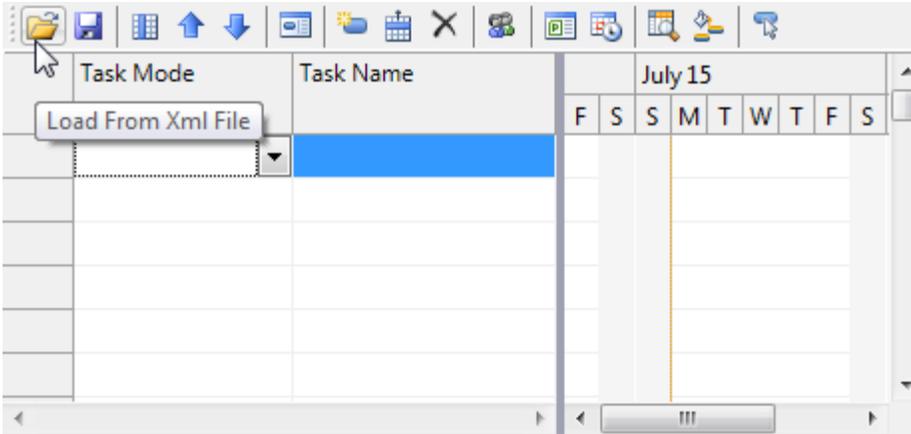
Loading GanttView From an XML File

This task shows how to load the **C1GanttView** as an XML File at run time and in code.

Load C1GanttView as an XML file at run time

To load the **C1GanttView** as an XML file at run time, complete the following:

1. Click the **Load From XML File** icon in the **C1GanttView** toolbar.



The **Load From Xml File** dialog box appears.

2. Browse to the location you wish to load the xml file.
3. Click Open in the **Load From Xml File** dialog box.

Load C1GanttView from XML file in code

To load the C1GanttView as an XML file in code, complete the following:

- **C#**

```
private void btnLoadXml_Click(object sender, EventArgs e)
{
    using (OpenFileDialog dlg = new OpenFileDialog())
    {
        dlg.DefaultExt = ".xml";
        dlg.Filter = "XML files|*.xml|All files|*.*";
        dlg.Title = "Load Gantt View From Xml File";
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            try
            {
                ganttView.LoadXml(dlg.FileName);
            }
            catch
            {
                MessageBox.Show("Bad C1GanttView XML.", dlg.Title);
            }
        }
    }
}
```

- **Visual Basic**

```
Private Sub btnLoadXml_Click(sender As Object, e As EventArgs)
    Using dlg As New OpenFileDialog()
        dlg.DefaultExt = ".xml"
        dlg.Filter = "XML files|*.xml|All files|*.*"
        dlg.Title = "Load Gantt View From Xml File"
        If dlg.ShowDialog() = DialogResult.OK Then
            Try
                ganttView.LoadXml(dlg.FileName)
            Catch
                MessageBox.Show("Bad C1GanttView XML.", dlg.Title)
            End Try
        End If
    End Using
End Sub
```

```
End If
End Using
End Sub
```

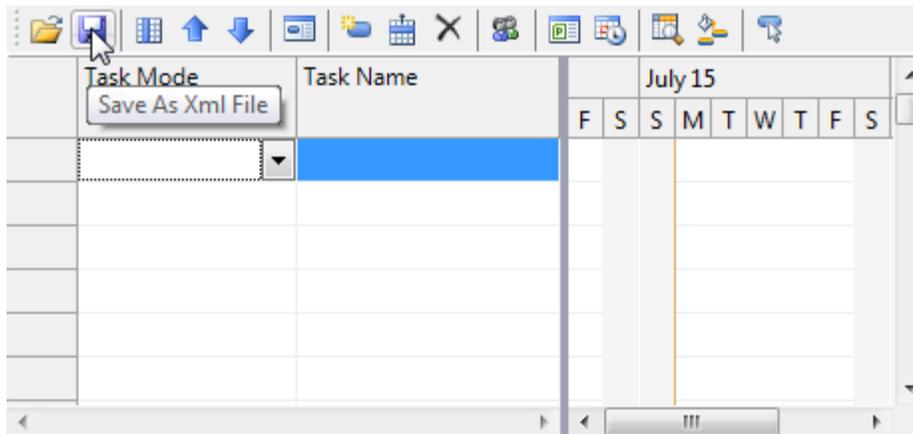
Saving GanttView as an XML File

This task shows how to save the C1GanttView as an XML File at run time and in code.

Save C1GanttView as an XML file at run time

To save the C1GanttView as an XML file at run time, complete the following:

1. Click the **Save as XML File** button from the C1GanttView toolbar button.



The **Save As Xml File** dialog box appears.

2. Browse to the location you wish to save the .xml file.
3. Click **Save** in the **Save As Xml File** dialog box.

Save C1GanttView from XML file in code

To save the C1GanttView as an XML file in code, complete the following:

- C#

```
private void btnSaveXml_Click(object sender, EventArgs e)
{
    using (SaveFileDialog dlg = new SaveFileDialog())
    {
        dlg.DefaultExt = ".xml";
        dlg.FileName = "gantt";
        dlg.Filter = "XML files|*.xml|All files|*.*";
        dlg.Title = "Save Gantt View As Xml File";
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            ganttView.SaveXml(dlg.FileName);
        }
    }
}
```

- Visual Basic

```
Private Sub btnSaveXml_Click(sender As Object, e As EventArgs)
    Using dlg As New SaveFileDialog()
```

```

        dlg.DefaultExt = ".xml"
        dlg.FileName = "gantt"
        dlg.Filter = "XML files|*.xml|All files|*.*"
        dlg.Title = "Save Gantt View As Xml File"
        If dlg.ShowDialog() = DialogResult.OK Then
            ganttView.SaveXml(dlg.FileName)
        End If
    End Using
End Sub

```

Customizing the Bar Style

To call attention to task bars on a Gantt view, such as a milestone, you can change their color, shape, or pattern to separate them from other bars of a particular type.

You could customize the appearance of all the bar styles or you could customize the appearance of an individual Gantt bar if you want to highlight a specific task in your plan.

Change the bar style at design time

1. Right-click on the **C1GanttView** control and select **Edit Bar Styles**.
The **C1GanttView.BarStyles Collection Editor** appears.
2. Click **Add** to add a bar style to the collection.
3. Set the **BarType** to **AutoTask**.
4. Set the **BarShape** to **ThickBar**.
5. Set the **BarColor** to **LightSkyBlue**.
6. Click **Add** to add a bar style to the collection.
7. Set the **BarType** to **ManualTask**.
8. Set the **BarShape** to **ThickBar**.
9. Set the **BarColor** to **PaleGreen**.
10. Click **OK** to save and close the **C1GanttView.BarStyles Collection Editor**.

Change the style of a specific task at design time

1. Right-click on the control and select **Edit Tasks**.
The **C1GanttView.Tasks Collection Editor** appears.
2. Select the task you want to customize from the members list and click on the ellipsis button next to the **BarStyles**.
The **C1GanttView.BarStyles Collection Editor** appears.
3. Click **Add** to add a bar style to the collection.
4. Set the **BarType** to **AutoTask**.
5. Set the **BarShape** to **TopBar**.
6. Set the **StartShape** and **EndShape** to **2**.
7. Set **RightText2** to **ResourceNames**.
8. Click **OK** to save and close the **C1GanttView.BarStyles Collection Editor**.

Change the bar style in code

To change the bar style for all manual tasks programmatically, complete the following:

- C#

```
private void btnChangeBarStyle_Click(object sender, EventArgs e)
{
    BarStyle bs = ganttView.GetPredefinedBarStyle(BarType.ManualTask);
    bs.BarColor = Color.LightCoral;
    ganttView.BarStyles.Add(bs);
}
```

- Visual Basic

```
Private Sub btnChangeBarStyle_Click(sender As Object, e As EventArgs)
    Dim bs As BarStyle =
ganttView.GetPredefinedBarStyle(BarType.ManualTask)
    bs.BarColor = Color.LightCoral
    ganttView.BarStyles.Add(bs)
End Sub
```

Change the style of a specific task in code

To change bar style for task3 programmatically, complete the following:

- C#

```
private void btnChangeTaskStyle_Click(object sender, EventArgs e)
{
    Task task3 = ganttView.Tasks.Search("Task 3");
    if (task3 != null)
    {
        BarStyle bs = ganttView.GetPredefinedBarStyle(BarType.ManualTask);
        bs.BarColor = Color.Green;
        bs.BarShape = BarShape.MiddleBar;
        bs.StartShape = 19;
        bs.EndShape = 19;
        task3.BarStyles.Add(bs);
    }
}
```

- Visual Basic

```
Private Sub btnChangeTaskStyle_Click(sender As Object, e As EventArgs)
    Dim task3 As Task = ganttView.Tasks.Search("Task 3")
    If task3 IsNot Nothing Then
        Dim bs As BarStyle =
ganttView.GetPredefinedBarStyle(BarType.ManualTask)
        bs.BarColor = Color.Green
        bs.BarShape = BarShape.MiddleBar
        bs.StartShape = 19
        bs.EndShape = 19
        task3.BarStyles.Add(bs)
    End If
End Sub
```

Creating a Custom Column

This topic shows how to programmatically create a custom column.

- C#

```
private void btnAddCustomColumn_Click(object sender, EventArgs e)
{
    CustomFieldColumn cc = new CustomFieldColumn();
    cc.Caption = "My Numeric Column";
    cc.DataType = typeof(decimal);
}
```

```

cc.Format = "$#0";
cc.Name = "MyNumericColumn";
cc.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
cc.Width = 65;
ganttView.Columns.Add(cc);
}

```

- Visual Basic

```

Private Sub btnAddCustomColumn_Click(sender As Object, e As EventArgs)

    Dim cc As New CustomFieldColumn()
    cc.Caption = "My Numeric Column"
    cc.DataType = GetType(Decimal)
    cc.Format = "$#0"
    cc.Name = "MyNumericColumn"
    cc.TextAlign = System.Windows.Forms.HorizontalAlignment.Right
    cc.Width = 65
    ganttView.Columns.Add(cc)

End Sub

```

Showing the Duration Columns in the Grid

This topic shows how to programmatically show/hide the values of the **Duration** and **DurationUnits** properties in the grid.

- C#

```

private void chkShowDuration_CheckedChanged(object sender, EventArgs e)
{
    TaskPropertyColumn durationCol =
ganttView.Columns.Search(TaskProperty.Duration);
    TaskPropertyColumn unitsCol =
ganttView.Columns.Search(TaskProperty.DurationUnits);
    if (durationCol != null && unitsCol != null)
    {
        bool visible = chkShowDuration.Checked;
        durationCol.Visible = visible;
        unitsCol.Visible = visible;
    }
}

```

- Visual Basic

```

Private Sub chkShowDuration_CheckedChanged(sender As Object, e As
EventArgs)
    Dim durationCol As TaskPropertyColumn =
ganttView.Columns.Search(TaskProperty.Duration)
    Dim unitsCol As TaskPropertyColumn =
ganttView.Columns.Search(TaskProperty.DurationUnits)
    If durationCol IsNot Nothing AndAlso unitsCol IsNot Nothing Then
        Dim visible As Boolean = chkShowDuration.Checked
        durationCol.Visible = visible
        unitsCol.Visible = visible
    End If
End Sub

```

Changing the Month the Fiscal Year Starts On

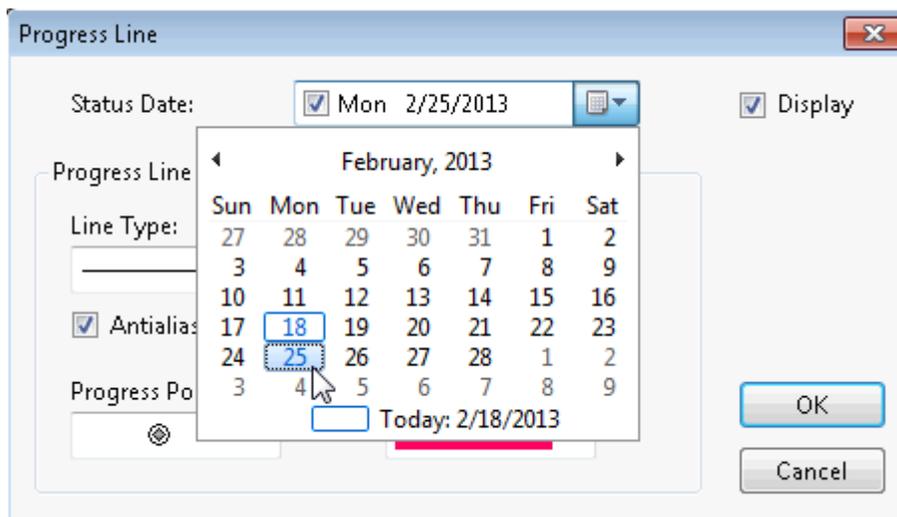
To change the month that the fiscal starts on at run time, complete the following:

1. Click the **Project Information** button from the **CI GanttView** toolbar button. The **Project Information** dialog box appears.
2. Select the **Calendar Options** tab.
3. Select **April** from the **Fiscal Year Starts in:** dropdown listbox.

Modifying the Progress Lines in your Project

To modify the progress lines on your timescale to represent your tasks's progress, complete the following:

1. On the **CI GanttView** toolbar, click the progress line button, . The **Progress Line** dialog box appears.
2. Click on the dropdown arrow next to the **Status Date** and select a date to indicate where the line is drawn. For example, select **Feb. 25, 2013** from the dropdown calendar.



3. Click on the dropdown arrow next to **Line Type** to specify the line type, for example select the dotted lines.
4. Click on the dropdown arrow next to the **Line Color** to specify a new line color for the progress lines. Select a color from the palette, for example, **Purple**.
5. Click on the dropdown arrow next to the **Progress Shape** and select the **Star**.
6. Click on the dropdown arrow next to the **Progress Point Color** and select **Yellow**.

This topic illustrates the following:

The Progress lines are drawn from Feb. 25, the given status date and the styles for the lines and points have been modified like the following:

