ComponentOne

GanttView for WinForms

ComponentOne, a division of GrapeCity
201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206 USA

Website: http://www.componentone.com
Sales: 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 media on which the software is delivered is 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 media 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 the defective media by sending it and a check for $2 5 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original media on which the software is delivered is 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. ComponentOne is 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.

Table of Contents

Copyright © 2017 GrapeCity, inc. All rights reserved.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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 ■ GanttView for WinForms Key Features ■ GanttView for WinForms Task-Based Help

Help with WinForms Edition

Getting Started

For information on installing **ComponentOne Studio WinForms Edition**, licensing, technical support, namespaces and creating a project with the control, please visit Getting Started with WinForms Edition.

Copyright © 2017 GrapeCity, inc. All rights reserved.

GanttView for WinForms Key Features

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

Automatic and Manual Scheduling

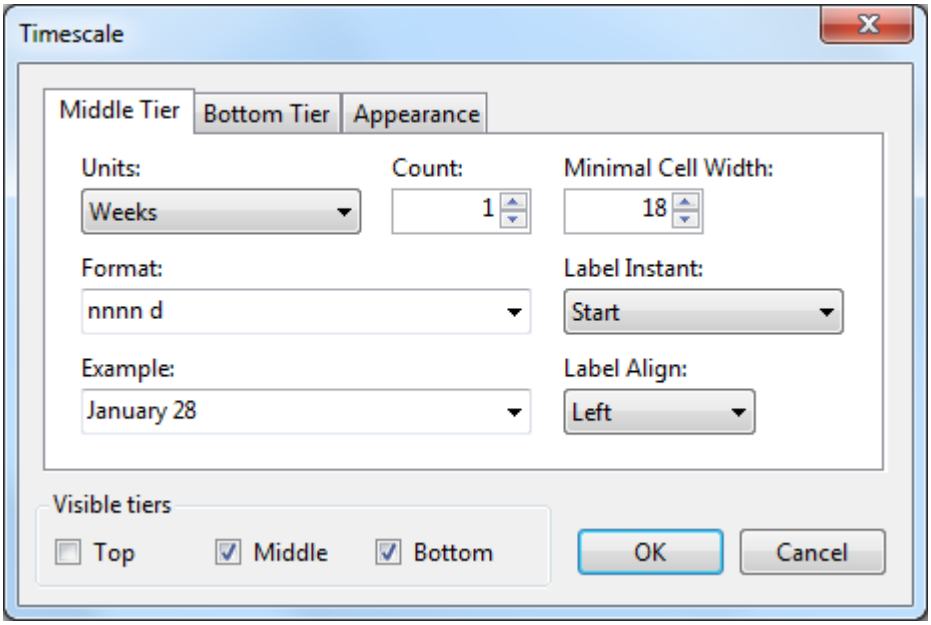
The C1GanttView control supports both Automatic and Manual scheduling modes like Microsoft Project. Automatic scheduling provides a highly structured, systematic means of managing project schedules. The project can be scheduled either from start or from the finish date. When you enter a start date, C1GanttView 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.

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.

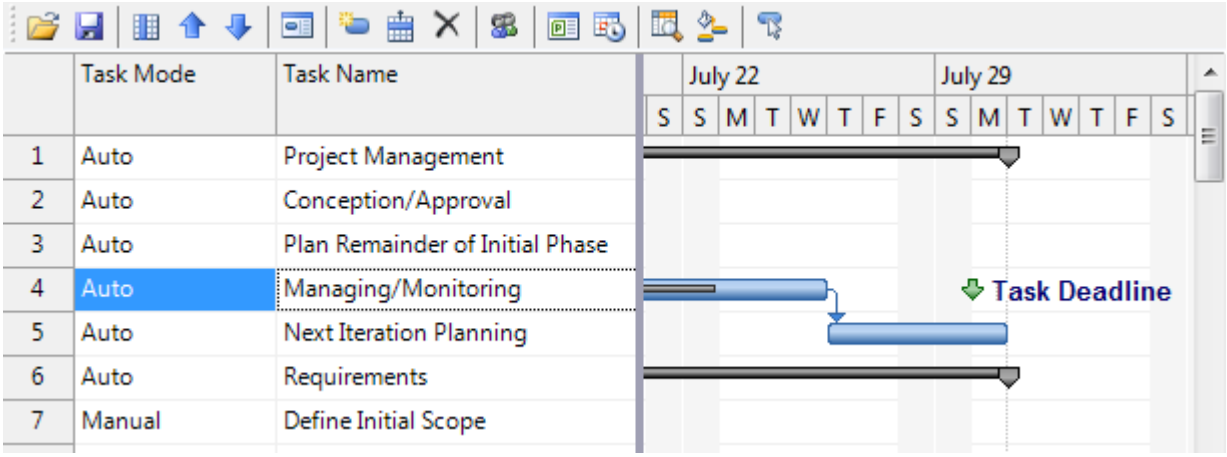
Copyright © 2017 GrapeCity, inc. All rights reserved.

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, or by dragging a task bar onto its successor. GanttView also supports validation to alert the user when scheduling conflicts arise.

Deadlines

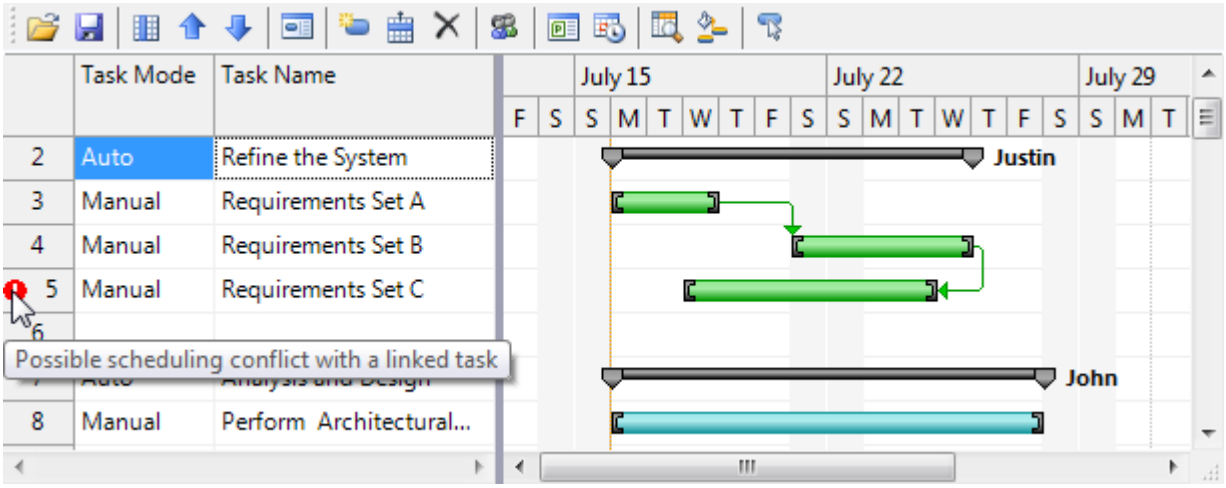
Set the deadline for any task to indicate when the task must be completed by. GanttView provides visual cues once a deadline is set. When a deadline is set for a specific task, a downward green arrow is drawn on the chart.



For conceptual information see, Task Deadline.

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

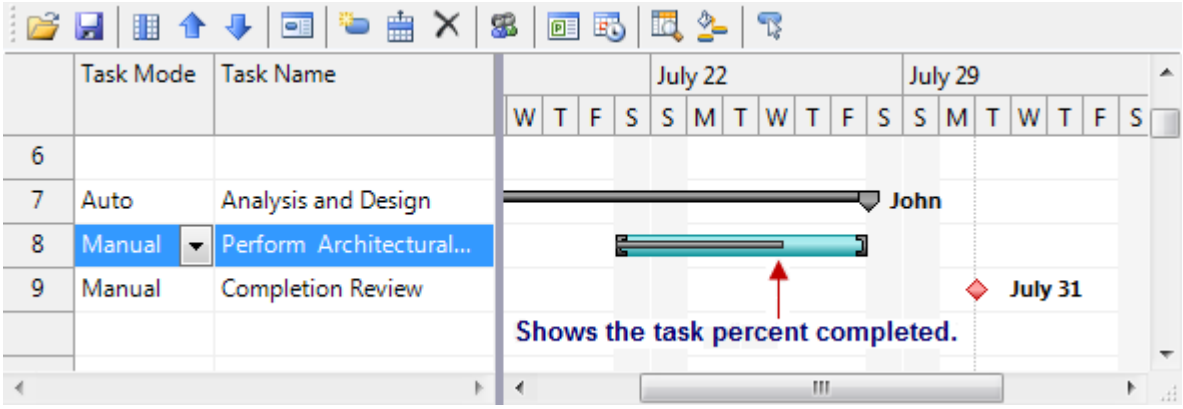
Copyright © 2017 GrapeCity, inc. All rights reserved.

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. For procedural information see, Assigning Resources to a Task.

Percent Complete

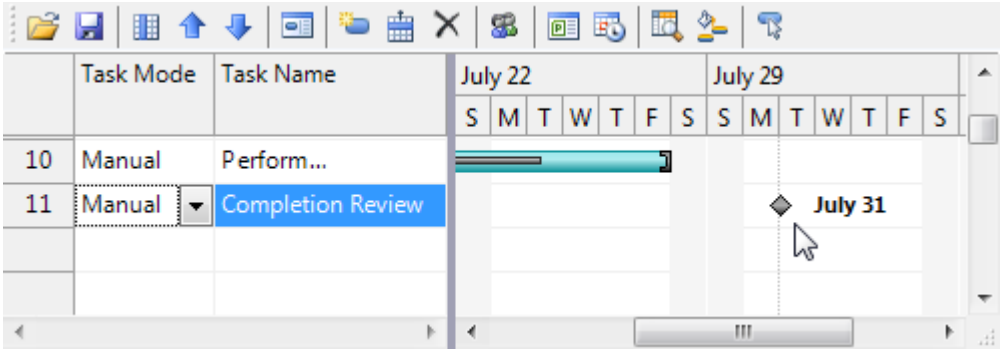
The task bars of the C1GanttView control can display like a progress bar to visualize task percentage complete.



For more information see, Task Percent Complete.

Milestones

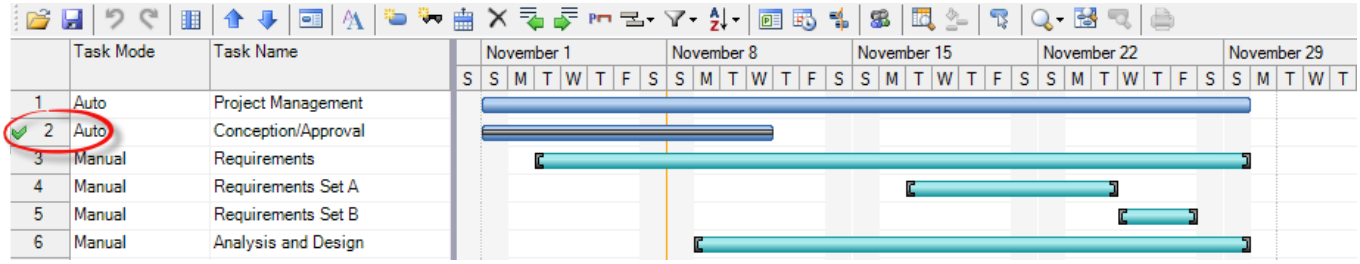
A milestone is a significant point or landmark in your project. Milestones are created as tasks with zero duration and are visualized with a diamond shape.



Completion Indicator

Completion indicator makes the tasks with 100% complete status stand out from the other tasks. This symbol appears in the grid column towards extreme left corresponding to the completed task.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

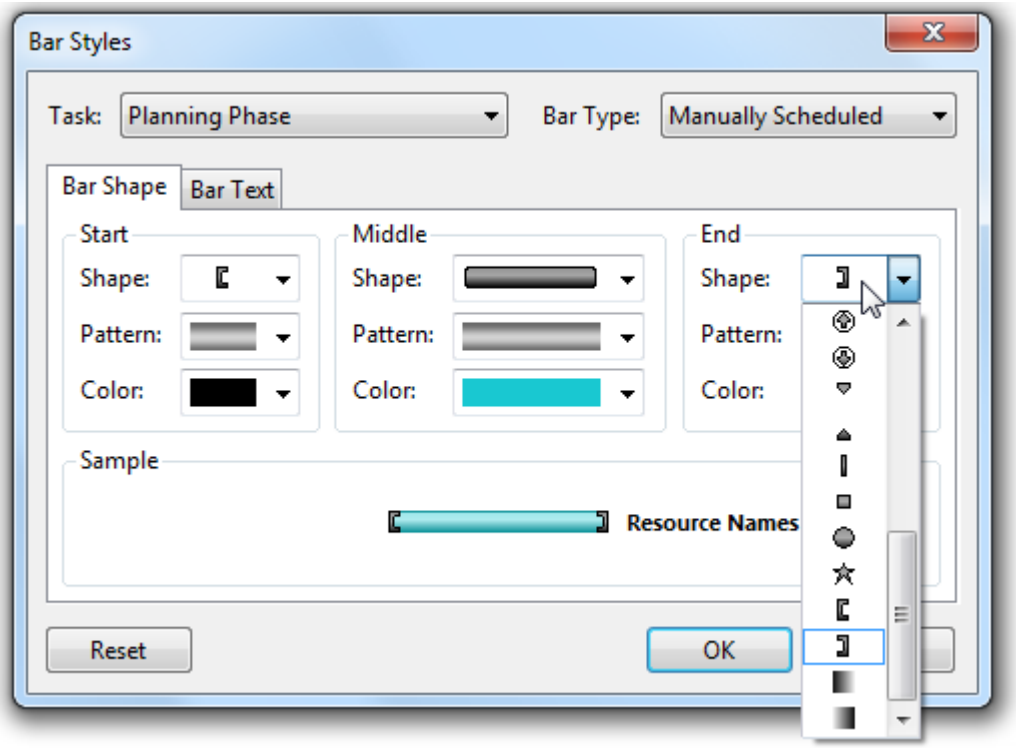
Custom Columns

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.

For more information see, Grid Columns Dialog Box.

Customizable Bar Styles

GanttView includes default styles for several different bar types including Auto Scheduled, Manually Scheduled, Progress Bar, Milestone, Deadline, etc. Users can customize the shape, pattern and color for each of these bar styles, or simply customize the style for one particular task. Users can also specify which text field to display on either side of the bar, such as Resource Names.

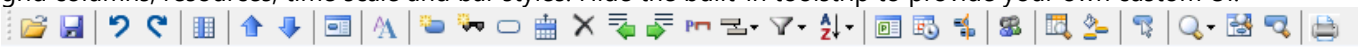


For more information on customizing the bar styles see, Customizing the Bar Style.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Built-in Toolstrip Commands

The **C1GanttView** control comes with a built-in toolstrip that contains the most common commands. Users can manage project information as well as move, insert and delete tasks in one click. Users can also manage the visible grid columns, resources, time scale and bar styles. Hide the built-in toolstrip to provide your own custom UI.



For more information see, C1GanttView Toolbar.

Import and Export to Microsoft Project

C1GanttView supports exporting of GanttView schedules to MS Project, and importing of MS Project xml to GanttView by using ExportToMsProjectXml and ImportFromMsProjectXml.

Copyright © 2017 GrapeCity, inc. All rights reserved.

GanttView for WinForms Quick Start

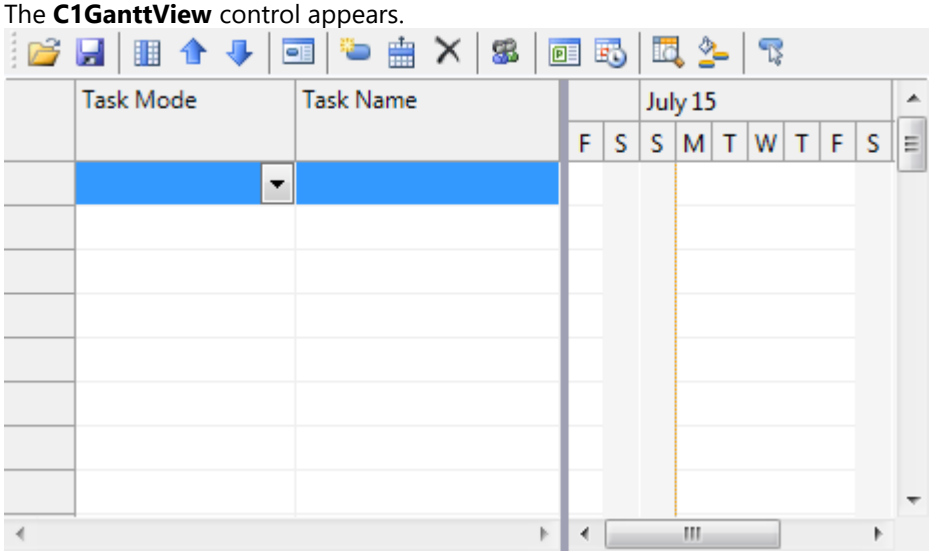
The goal of this quick start guide is to get you acquainted with **GanttView for WinForms**. In the first step of this Quick Start guide, you will add a C1GanttView control to your WinForms project. This quick start guide will also explain how to add tasks, resources, and other schedule information to 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 **C1GanttView.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.
4. Set **Days** to **30**.

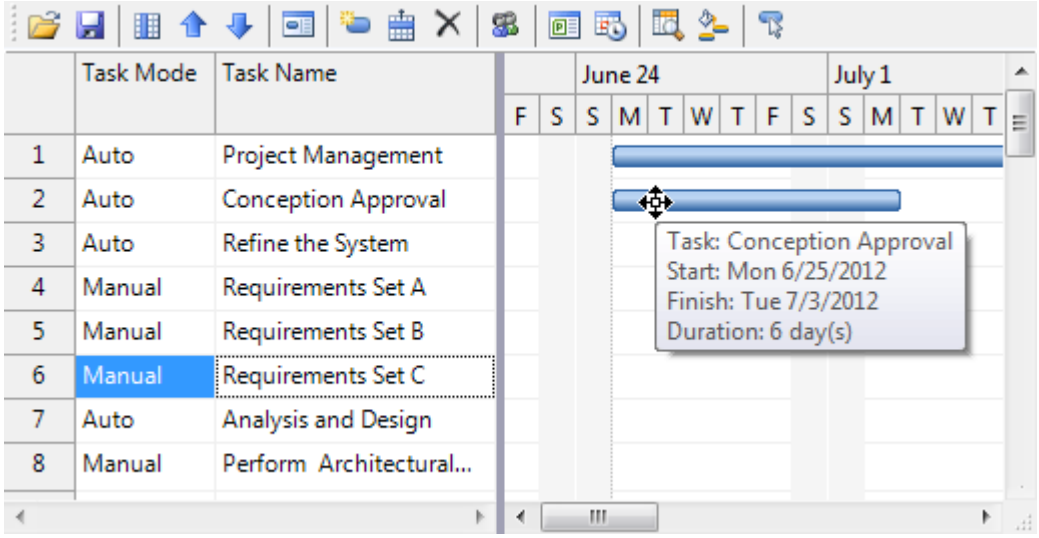
Copyright © 2017 GrapeCity, inc. All rights reserved.

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 **7/10/2012**.
7. Set the **Mode** to **Automatic**.
8. Set the **PercentComplete** to **100%**. For more information see Task Percent Complete.
9. Select the third member from the list, member[2] and enter a name next to the **Name** property such as, **Refine the System**.
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 **7/27/2012**.
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 **7/23/2012**.
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 **7/30/2012**.
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 **7/31/2012**.
22. Click **OK** to save and close the **C1GanttView.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 tasks 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 C1GanttView Appearance Properties.
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 C1GanttView Appearance Properties.
5. Hover over each task bar and notice a tooltip appears with the task's summary.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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 **C1GanttView.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 **C1GanttView.Resources Collection Editor**.

To assign resources to specific tasks at design time:

1. Right-click on the control and select **Edit Tasks**.
   The **C1GanttView.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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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. 10. Click **OK** to save and close the **Task.Predecessors Collection Editor**.
    For more information on predecessors see Creating Predecessors.

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.

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. 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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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.

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

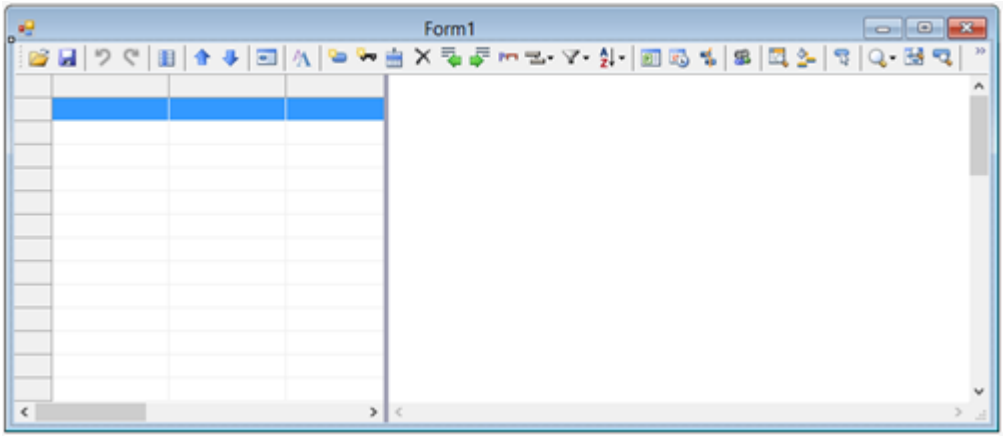
Data Binding: GanttView for WinForms

This section walks you through step-by-step instructions for achieving data binding in **GanttView for WinForms**. In this topic, you start by creating a new WinForms project in Visual Studio, add the GanttView control to the application, and bind the GanttView to a data source. In addition, you customize some of the features of the GanttView control at design time and run the application to see how data binding works in the GanttView control.

Step 1: Setting up the Application

In this step, you begin by creating a WinForms application in Visual Studio and then adding a GanttView control to your application.

1. Create a new WinForms project in Visual Studio.
2. Navigate to the Toolbox and double-click the C1GanttView icon to add the control onto the form. Note that the following references get added to the project.
   - C1.Win.C1GanttView.4
   - C1.Win.C1Schedule.4
   - C1.Win.FlexGrid.4
3. Resize the form and the GanttView control. Click the GanttView control once in design view and set the Dock property to **Fill** in the Properties window.
4. The design view would appear similar to the following image.



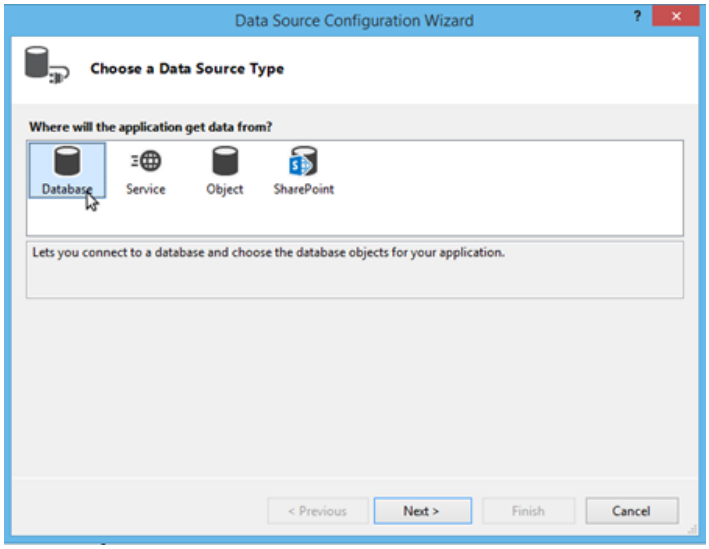
With this, you have set up the basic application to implement data binding in **GanttView for WinForms.**

Step 2: Configuring the Data Source

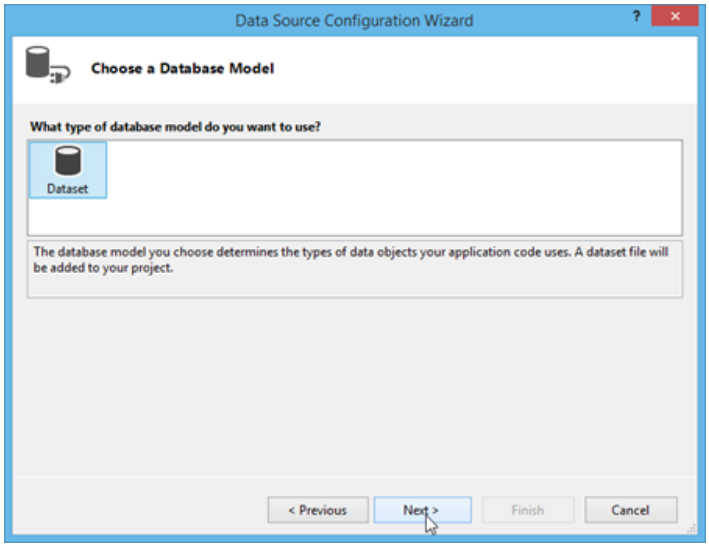
In this step, you start by configuring a data source to your project and then adding code to synchronize data with the GanttView control.

1. To add a data source to your project, click **View | Other Windows | Data Sources.**
2. In Data Sources window, click **Add New Data Source** link to open **Data Source Configuration Wizard** window. Select Database as the Data Source Type and then click Next as shown below.

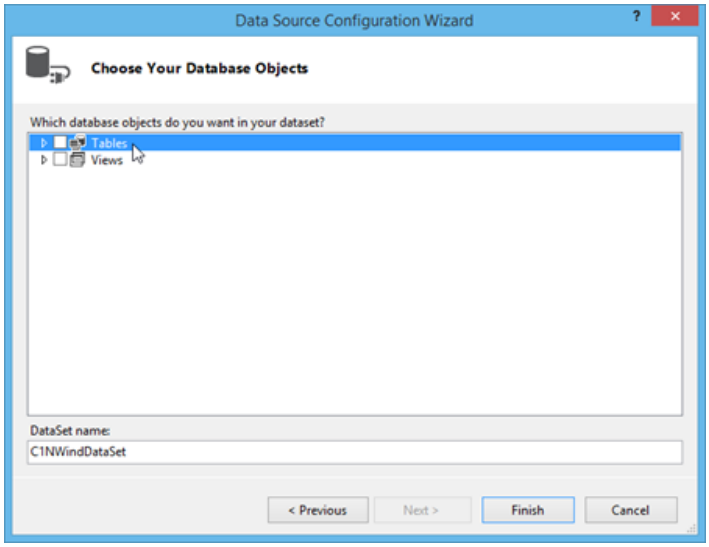
Copyright © 2017 GrapeCity, inc. All rights reserved.



3. After clicking Next, select **Dataset** as the database model in **Data Source Configuration Wizard** window and click Next as shown below.



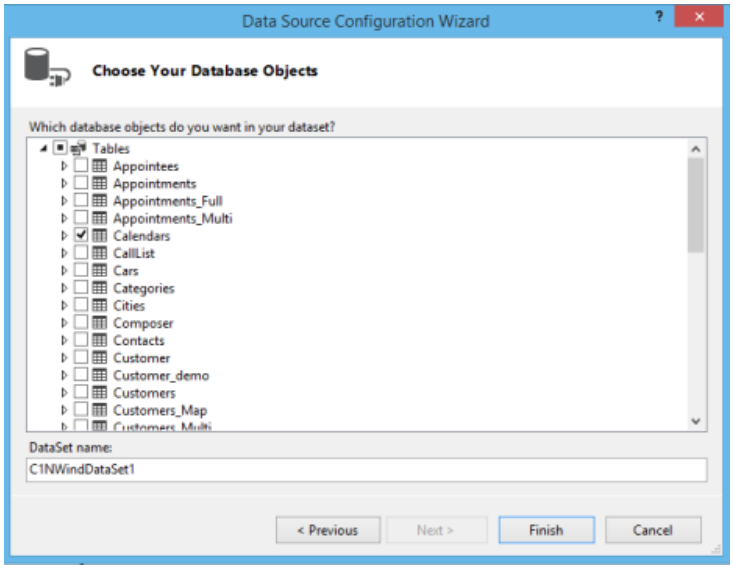
4. Select Data Connection by browsing **C1NWind.mdb** database file in your system. Test the connection and click Next. Save the connection string to add the database file to your project and click Next.
5. Choose **Tables** as Database Object Model in **Data Source Configuration Wizard** window as shown below.



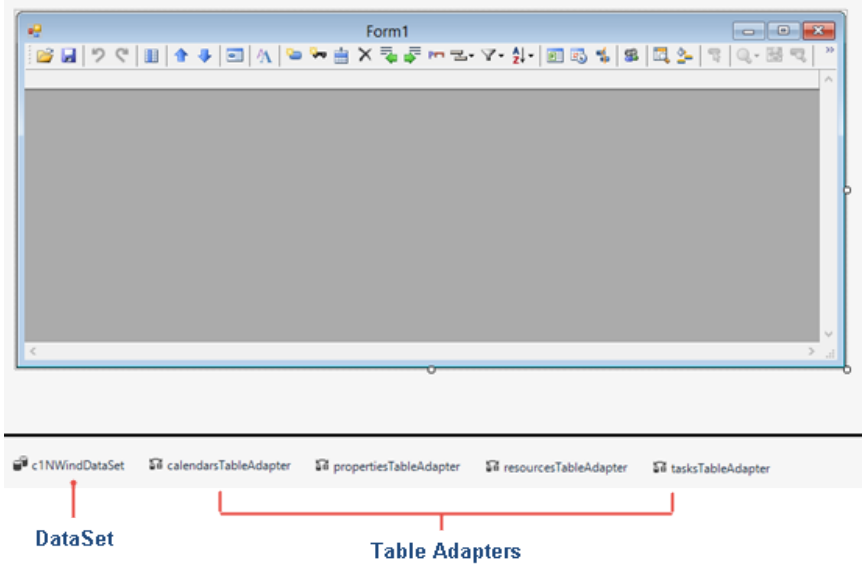
6. For this data binding topic, select **Calendars, Properties, Resources** and **Tasks** tables as object models from the drop-down list appearing

Copyright © 2017 GrapeCity, inc. All rights reserved.

under Tables and click Finish as shown below.



7. Navigate to the Toolbox again and locate the BindingSource icon in Data. Double-click the BindingSource icon to add it to the Design View.
8. Click the BindingSource once and set the DataSource property to **c1NWindDataSet** and DataMember as **Tasks** using the drop-down arrow in the Properties window**.** Delete the BindingSource from the design view after setting these properties.
9. Repeat **Step 8** for Calendars, Resources and Properties tables to set their DataSource as **c1NWindDataSet** and DataMember as **Calendars, Resources** and **Properties**, respectively.
10. The above steps creates adapters for the added datasets in your project. These adapters appear in design view similar to the following image.



11. Add **LoadData()** and **SaveData()** methods in the code to load and save data from the data source. These methods get invoked in **Form1_Load** and **Form1_FormClosing** events, respectively, as shown in the following code.

Visual Basic copyCode

```
Imports C1.Win.C1GanttView

Public Class Form1

    Public Sub New()
        InitializeComponent()
    End Sub

    Private Sub LoadData()
        Try
            Me.TasksTableAdapter.Fill(Me.C1NWindDataSet.Tasks)
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
            Me.ResourcesTableAdapter.Fill(Me.C1NWindDataSet.Resources)

            Me.PropertiesTableAdapter.Fill(Me.C1NWindDataSet.Properties)

            Me.CalendarsTableAdapter.Fill(Me.C1NWindDataSet.Calendars)

        Catch ex As Exception

            MessageBox.Show(ex.ToString)

        End Try
    End Sub

    Private Sub SaveData()
        Try
            Me.TasksTableAdapter.Update(C1NWindDataSet.Tasks)

            Me.ResourcesTableAdapter.Update(C1NWindDataSet.Resources)

            Me.PropertiesTableAdapter.Update(C1NWindDataSet.Properties)

            Me.CalendarsTableAdapter.Update(C1NWindDataSet.Calendars)

        Catch ex As Exception

        End Try
    End Sub



    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        LoadData()

    End Sub

    Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

        SaveData()

    End Sub
End Class
```

C# copyCode

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

        private void LoadData()
    {
        try
        {
        this.tasksTableAdapter.Fill(this.c1NWindDataSet.Tasks);
        this.resourcesTableAdapter.Fill(this.c1NWindDataSet.Resources);
        this.propertiesTableAdapter.Fill(this.c1NWindDataSet.Properties);
        this.calendarsTableAdapter.Fill(this.c1NWindDataSet.Calendars);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
    private void SaveData()
    {
        try
        {
            this.tasksTableAdapter.Update(c1NWindDataSet.Tasks);
            this.resourcesTableAdapter.Update(c1NWindDataSet.Resources);
            this.propertiesTableAdapter.Update(c1NWindDataSet.Properties);
            this.calendarsTableAdapter.Update(c1NWindDataSet.Calendars);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        LoadData();

    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        SaveData();
    }
    }
```

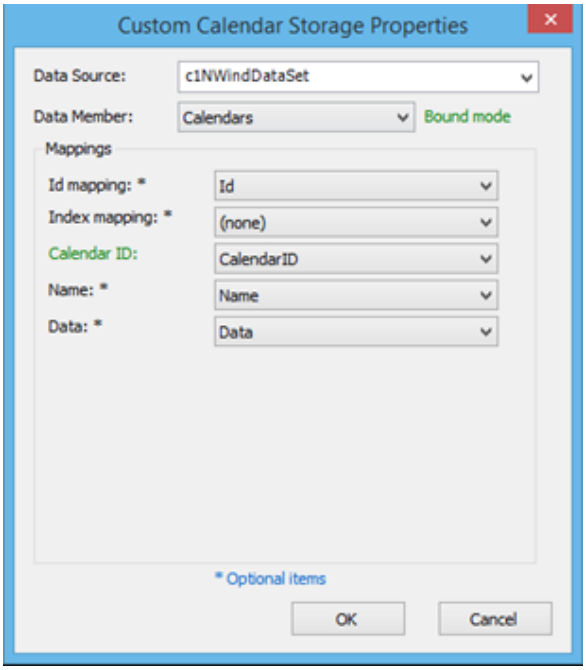
With this, you have configured C1NWind.mdb database to your project and added code to synchronize the data in both the directions.

Step 3: Binding GanttView to the Data Source

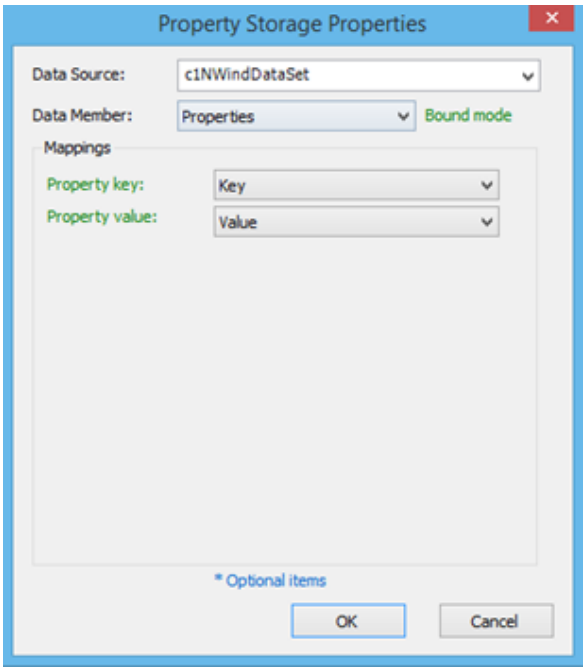
In this step, you begin by setting selected properties of C1GanttView class at design-time. These settings are to bind the GanttView control to the added data source, and enhance the visual appearance of the control.

1. In the form, select the GanttView control and navigate to the Properties window to make the following settings under **DataStorage** component.
    - **CalendarStorage:**
        - DataMember - Calendars
        - DataSource - c1NWindDataSet.
    - **PropertyStorage:**
        - DataMember - Properties
        - DataSource - c1NWindDataSet
        - Key MappingName - Key ; Value MappingName - Value.
    - **ResourceStorage:**
        - DataMember - Resources
        - DataSource - c1NWindDataSet
    - **TaskStorage:**
        - DataMember - Tasks
        - DataSource - c1NWindDataSet
2. In **DefaultWorkingTimes** component in Properties window, make the following settings.
    - **Interval_1:**
        - Empty - False
        - From - 9:00 AM.
        - To - 1:00 PM
    - **Interval_2:**
        - Empty - False
        - From - 2:00 PM.
        - To - 6:00 PM
3. Click the ellipsis appearing in the CalendarStorage property to set the following mappings and click OK.

Copyright © 2017 GrapeCity, inc. All rights reserved.

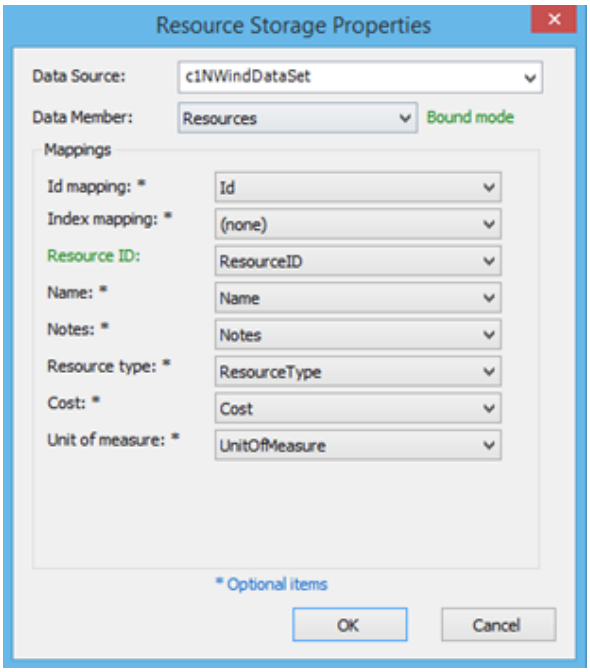


4. Click the ellipsis appearing in the PropertyStorage property to set the following mappings and click OK.

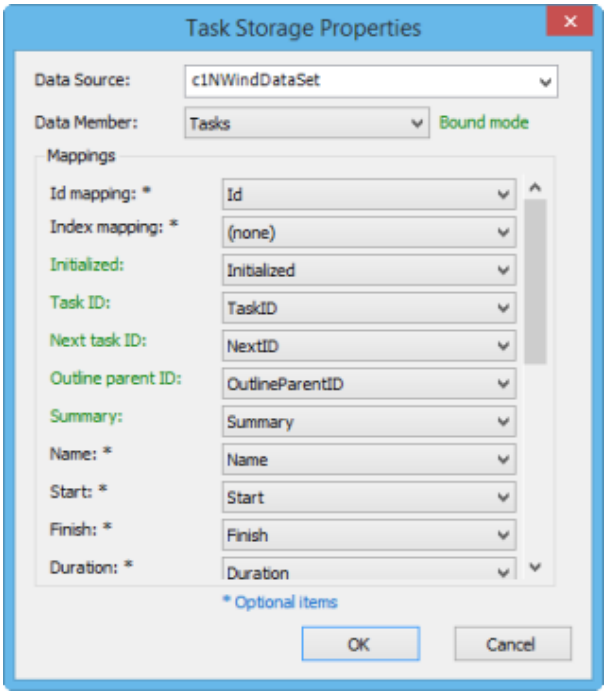


5. Click the ellipsis appearing in the ResourceStorage property to set the following mappings and click OK.

Copyright © 2017 GrapeCity, inc. All rights reserved.



6. Click the ellipsis appearing in the TasksStorage property to set the following mappings and click OK.



7. Refer to the following table to see all the mappings required for TasksStorage property.

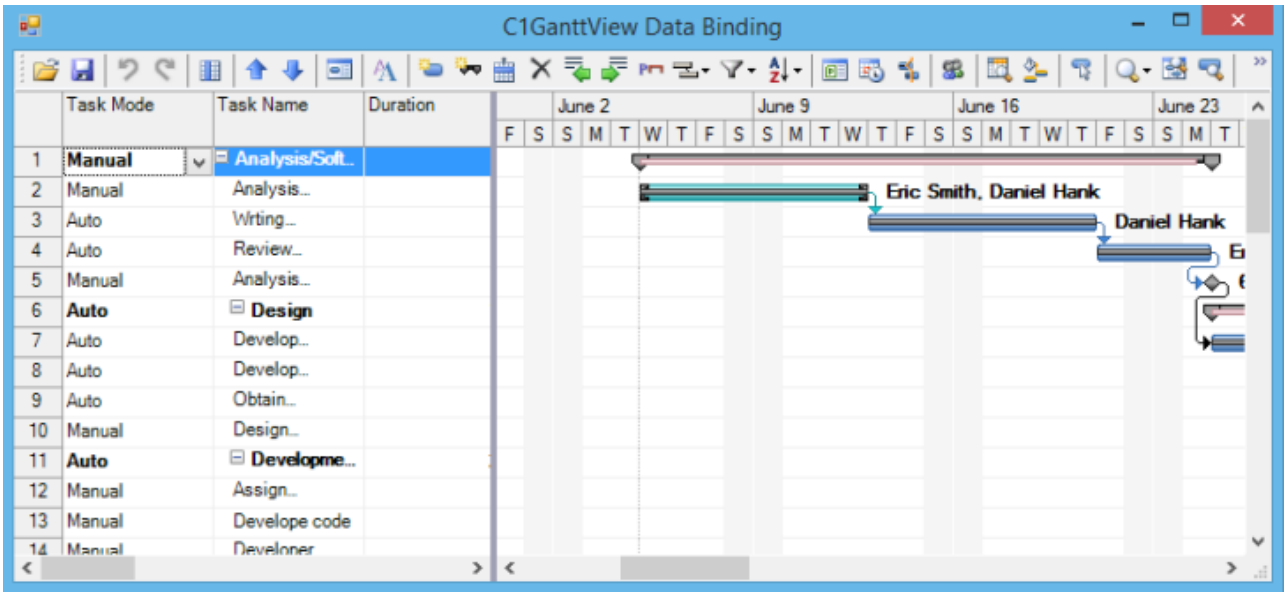
Id Mapping	Id
Initialized	Initialized
Task ID	TaskID
Next task ID	NexID
Outline parent ID	OutlineParentID
Summary	Summary
Name	Name

Copyright © 2017 GrapeCity, inc. All rights reserved.

Start	Start
Finish	Finish
Duration	Duration
Mode	Mode
Calendar ID	CalendarID
Duration units	DurationUnits
Percent complete	PercentComplete
Deadline	Deadline
Constraint type	ConstraintType
Hide bar	HideBar
Predecessors	Predecessors
Resources	Resources
Custom fields	CustomFields
Notes	Notes

8. In the Resources property, add four resources through **Resource Collection Editor**. Name these resources as Resource 1, Resource 2 and so on.
9. Locate the **TimeScale** component in Properties window to make the following timescale settings.
    - **MiddleTier:**
        - Units - Weeks
        - Count - 1
        - Format - nnnn d
        - Visible - True
    - **BottomTier:**
        - Units - Days
        - Count - 1
        - Format - w
        - Align - Center
        - Visible - True
        - UseFiscalYear - True
10. Press **F5** to run the application. The output window displays the GanttView control with data fetched from the added data source. The GanttView control appears similar to the following image.

Copyright © 2017 GrapeCity, inc. All rights reserved.



With this, you have completed data binding in **GanttView for WinForms**.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Design-Time Support

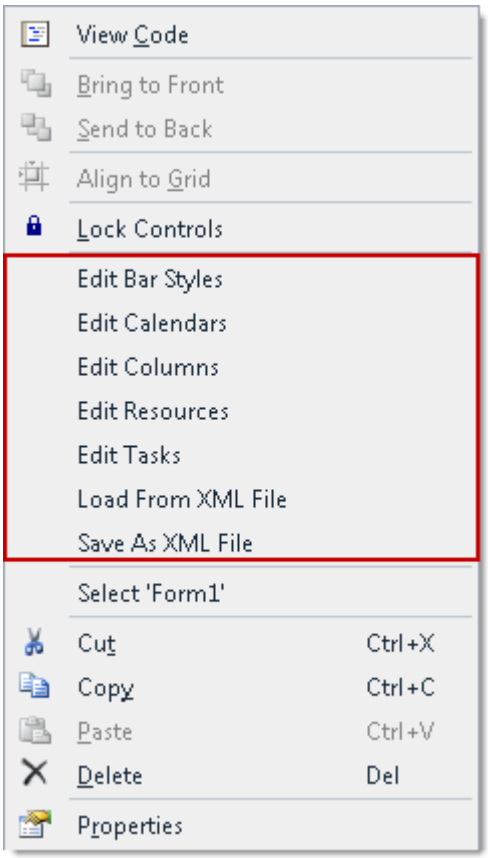
C1GanttView provides customized context menus, smart tags, and a designer that offers rich design-time support and simplifies working with the object model.

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,

Copyright © 2017 GrapeCity, inc. All rights reserved.

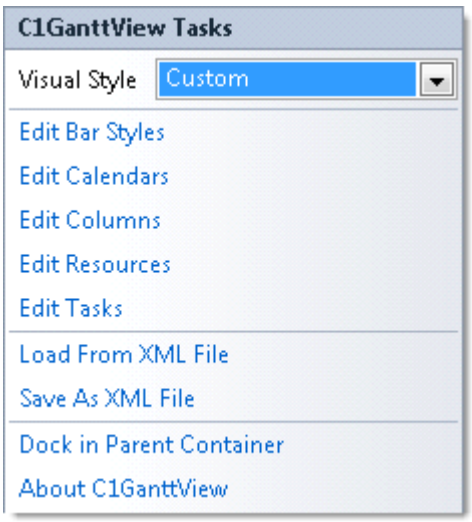
remove, or modify bar styles for the bars that represent the milestones.

- **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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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.

Dock in Parent Container

Clicking **Dock in Parent Container** allows the C1GanttView control to occupy the entire form.

About C1GanttView

Clicking **About C1GanttView** shows a dialog box. This dialog box displays the version number and licensing information for the C1GanttView product.

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:

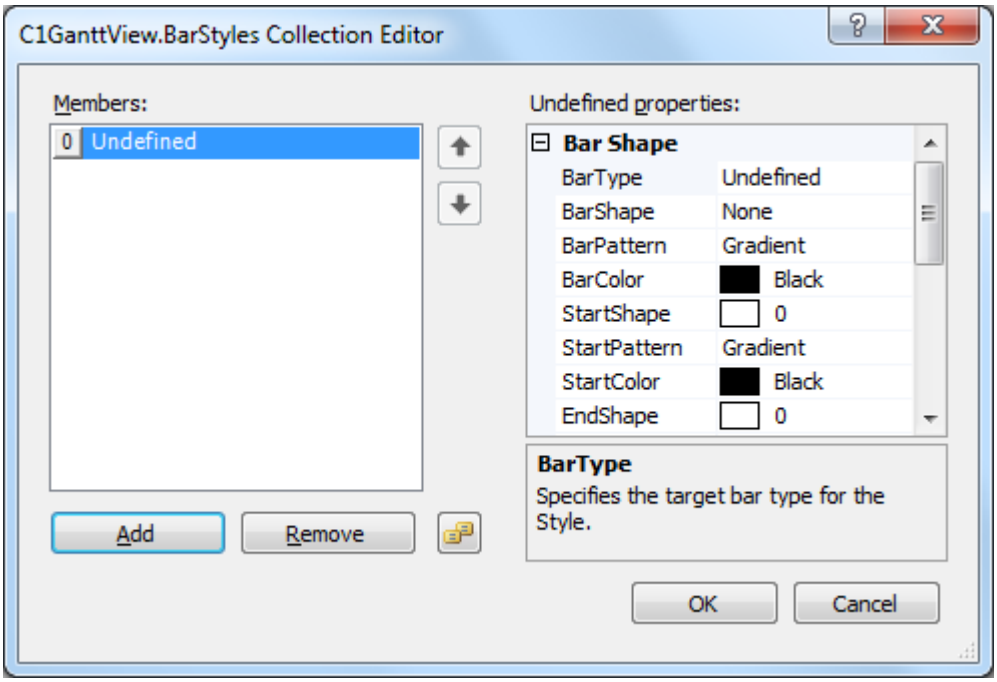
Copyright © 2017 GrapeCity, inc. All rights reserved.

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:



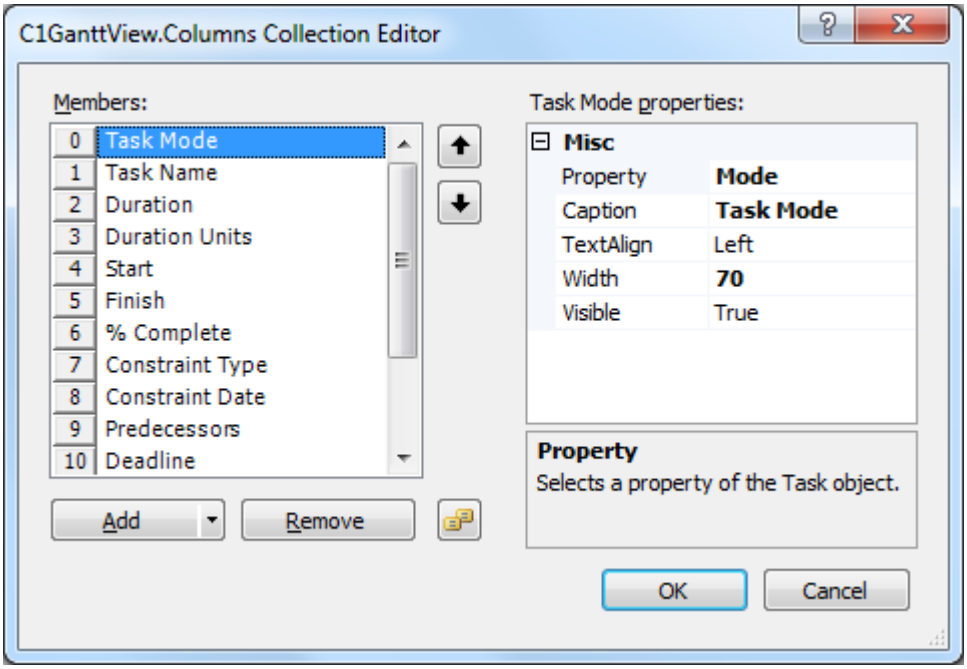
Columns Collection Editor

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:

Copyright © 2017 GrapeCity, inc. All rights reserved.

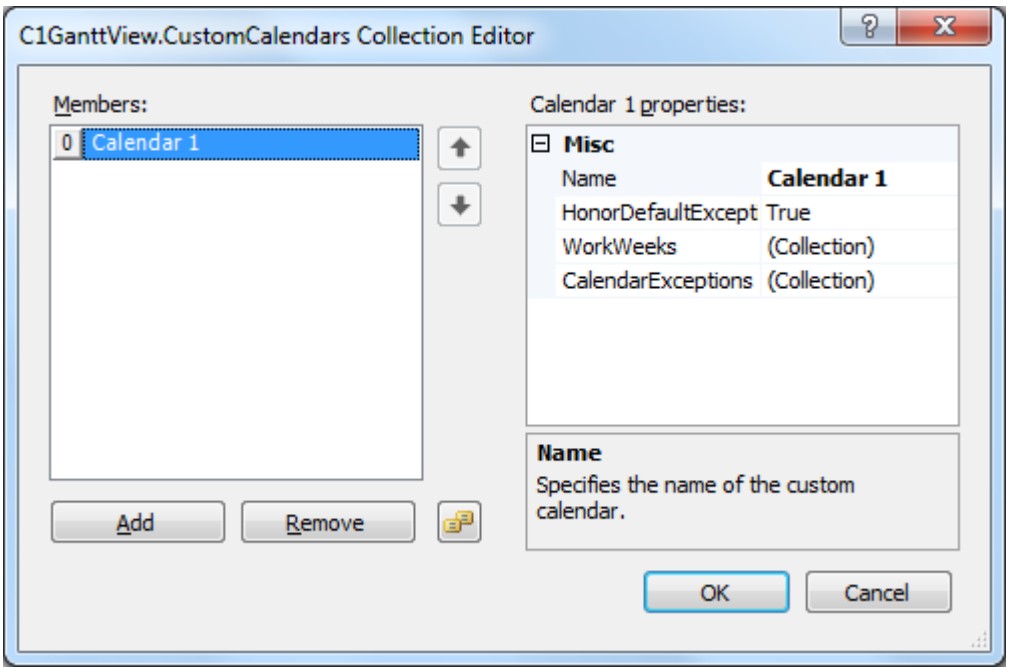


CustomCalendars Collection Editor

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:



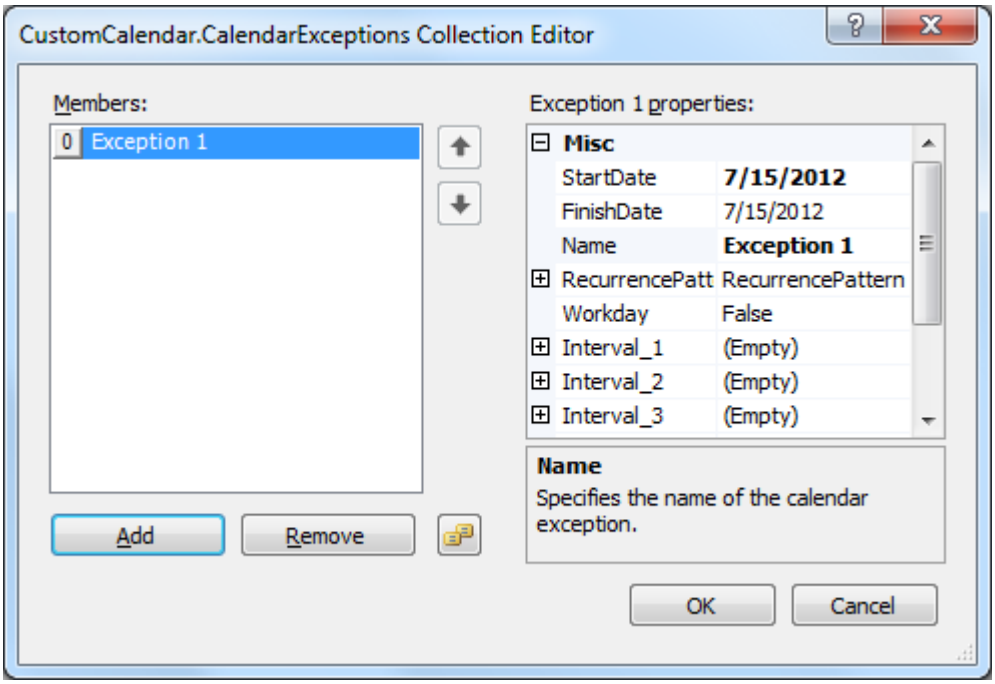
Copyright © 2017 GrapeCity, inc. All rights reserved.

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:



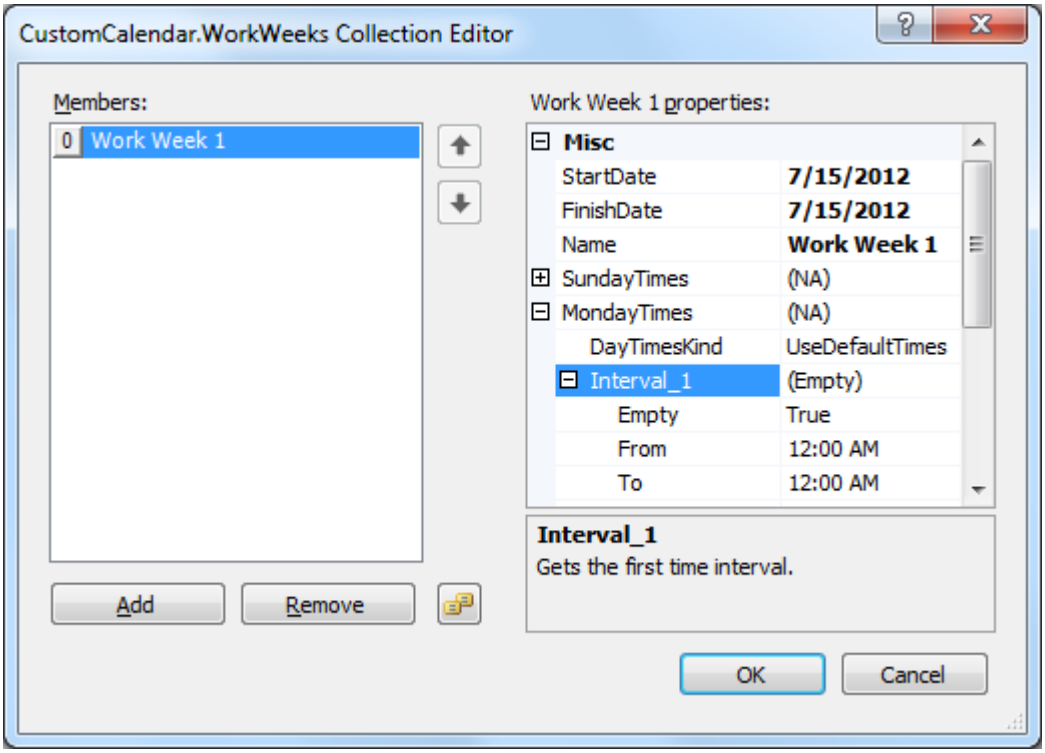
CustomCalendar WorkWeek Collection Editor

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:

Copyright © 2017 GrapeCity, inc. All rights reserved.

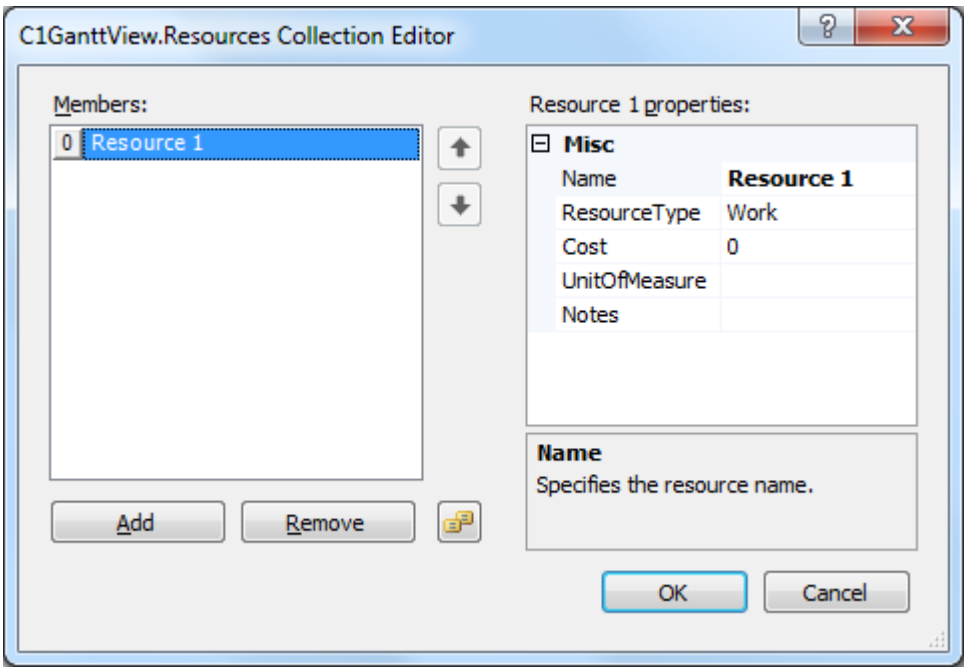


Resources Collection Editor

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:



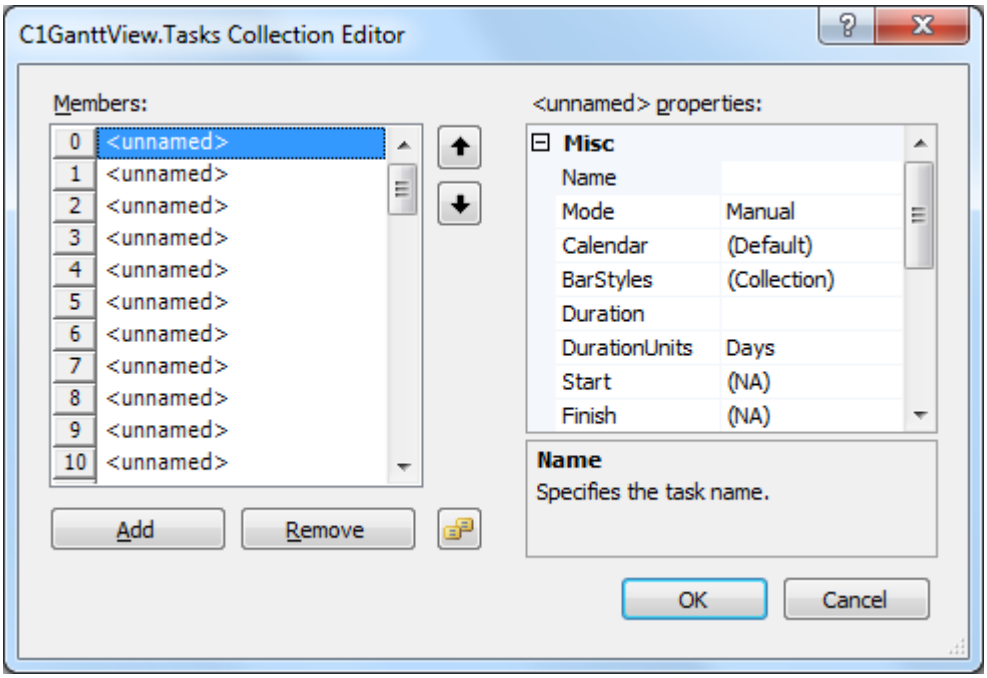
Copyright © 2017 GrapeCity, inc. All rights reserved.

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:



Copyright © 2017 GrapeCity, inc. All rights reserved.

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
- Change Working Time Dialog Box
- Grid Columns Dialog Box
- Page Setup Dialog Box
- Print Preview Dialog Box
- Progress Line Dialog Box
- Print Dialog Box
- Project Information Dialog Box
- Project Resources Dialog Box
- Style Settings Dialog Box
- Task Information Dialog Box
- Link Information Dialog Box
- Timescale Dialog Box
- Zoom Dialog Box
- Advanced Filter Dialog Box

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

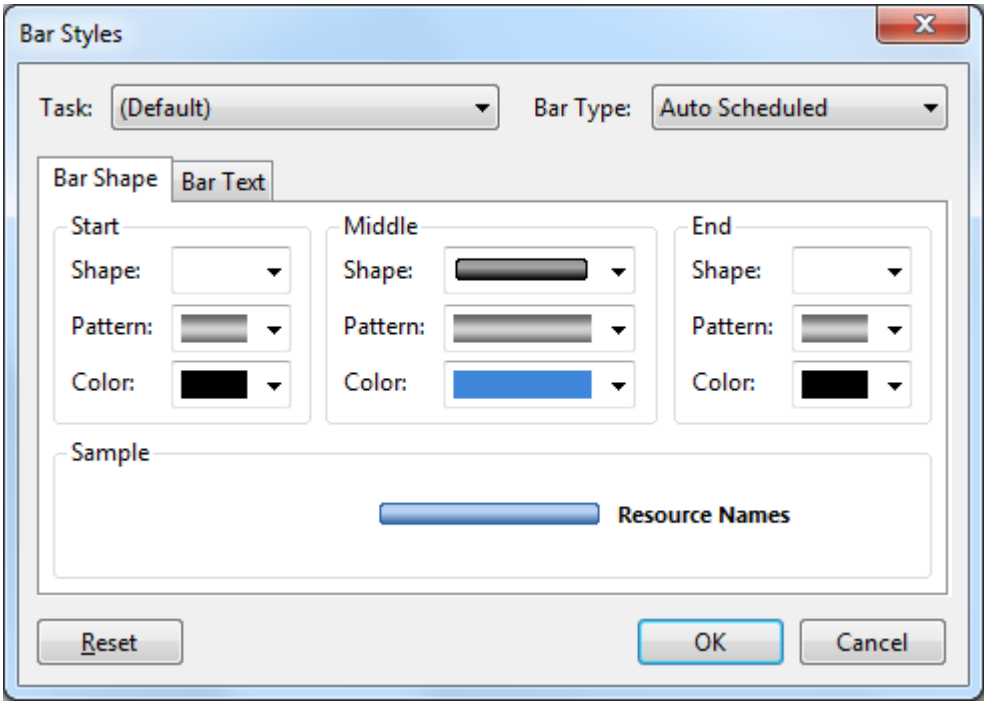
Bar Styles Dialog Box

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.



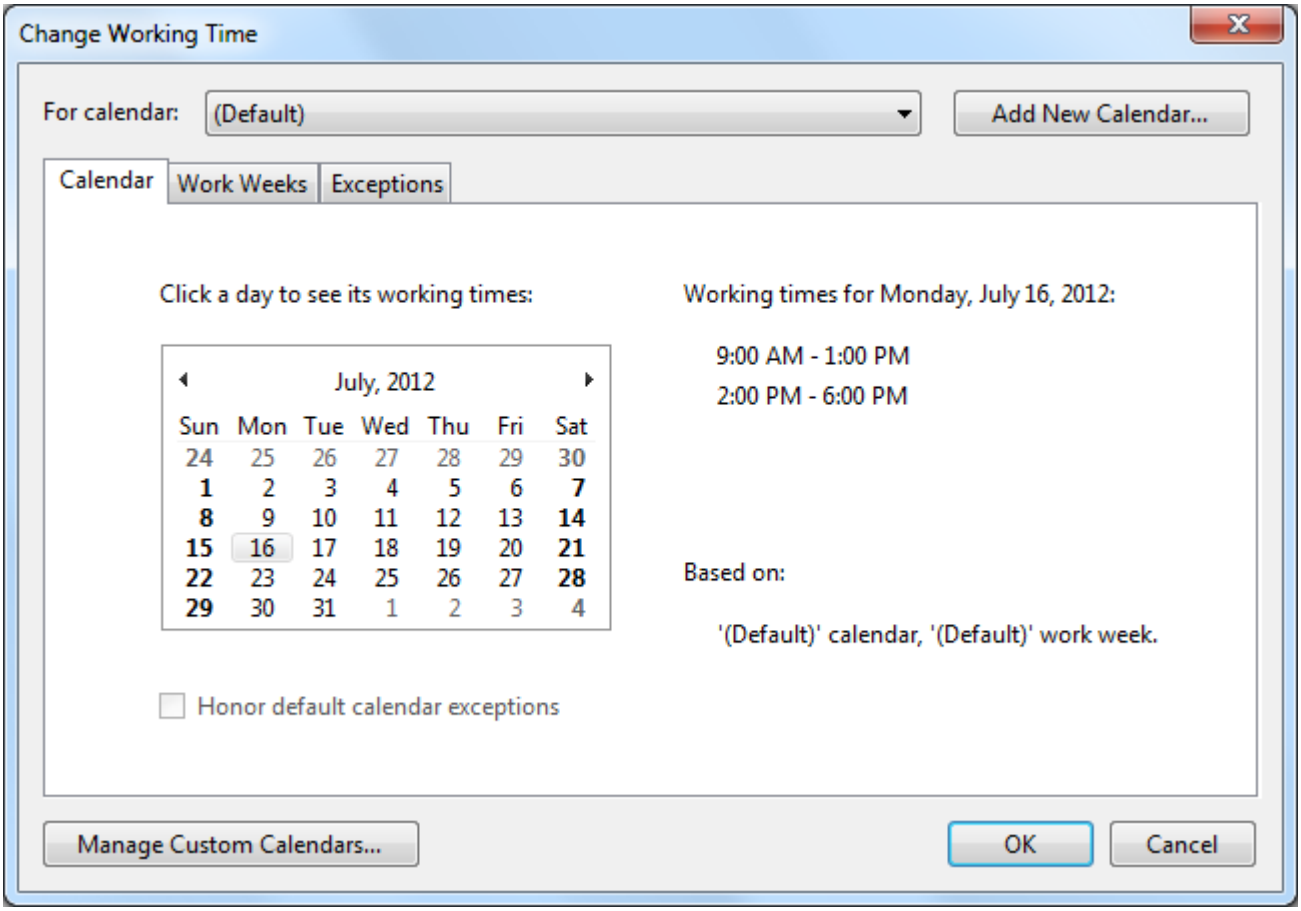
Change Working Time Dialog Box

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.

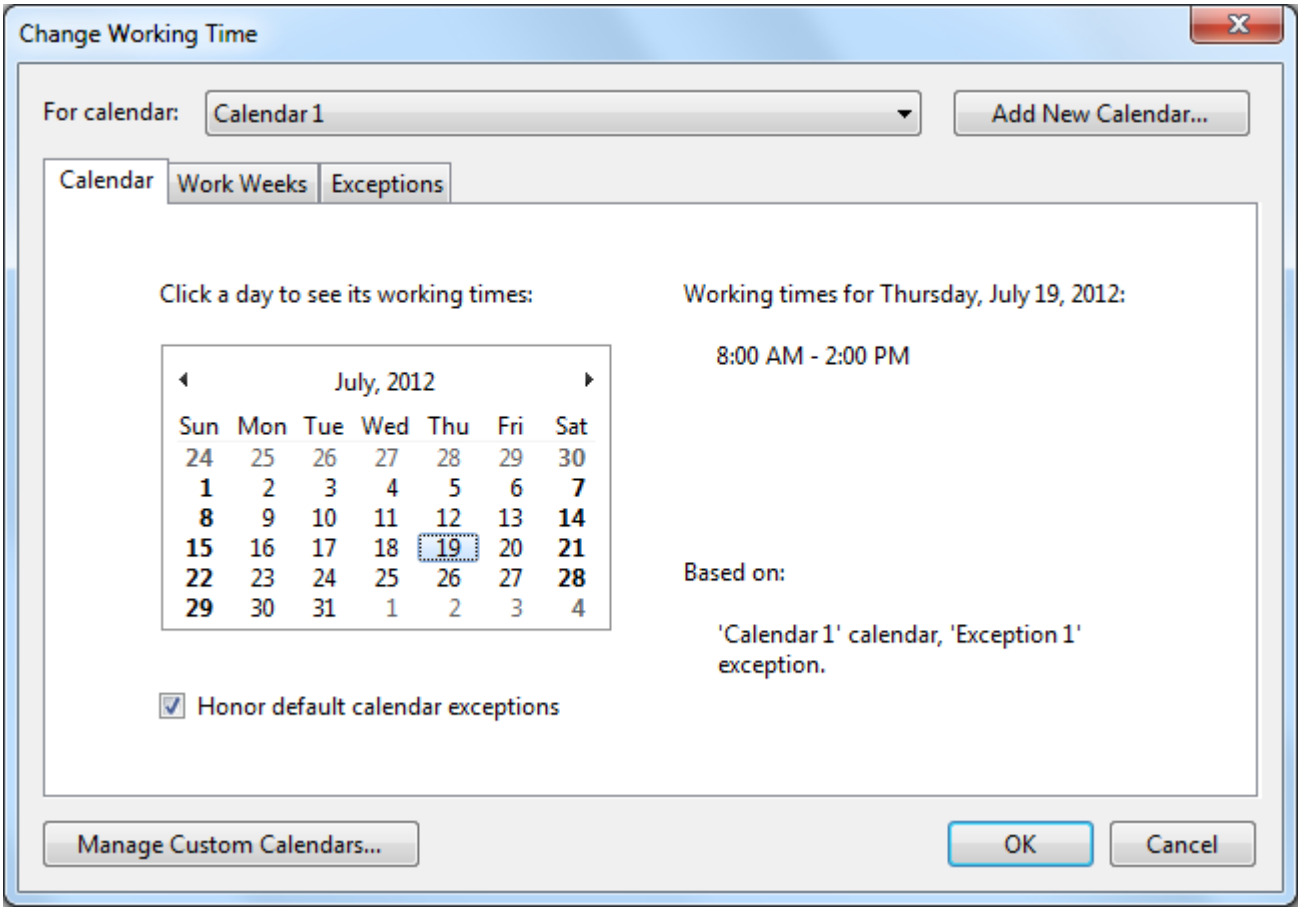
Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

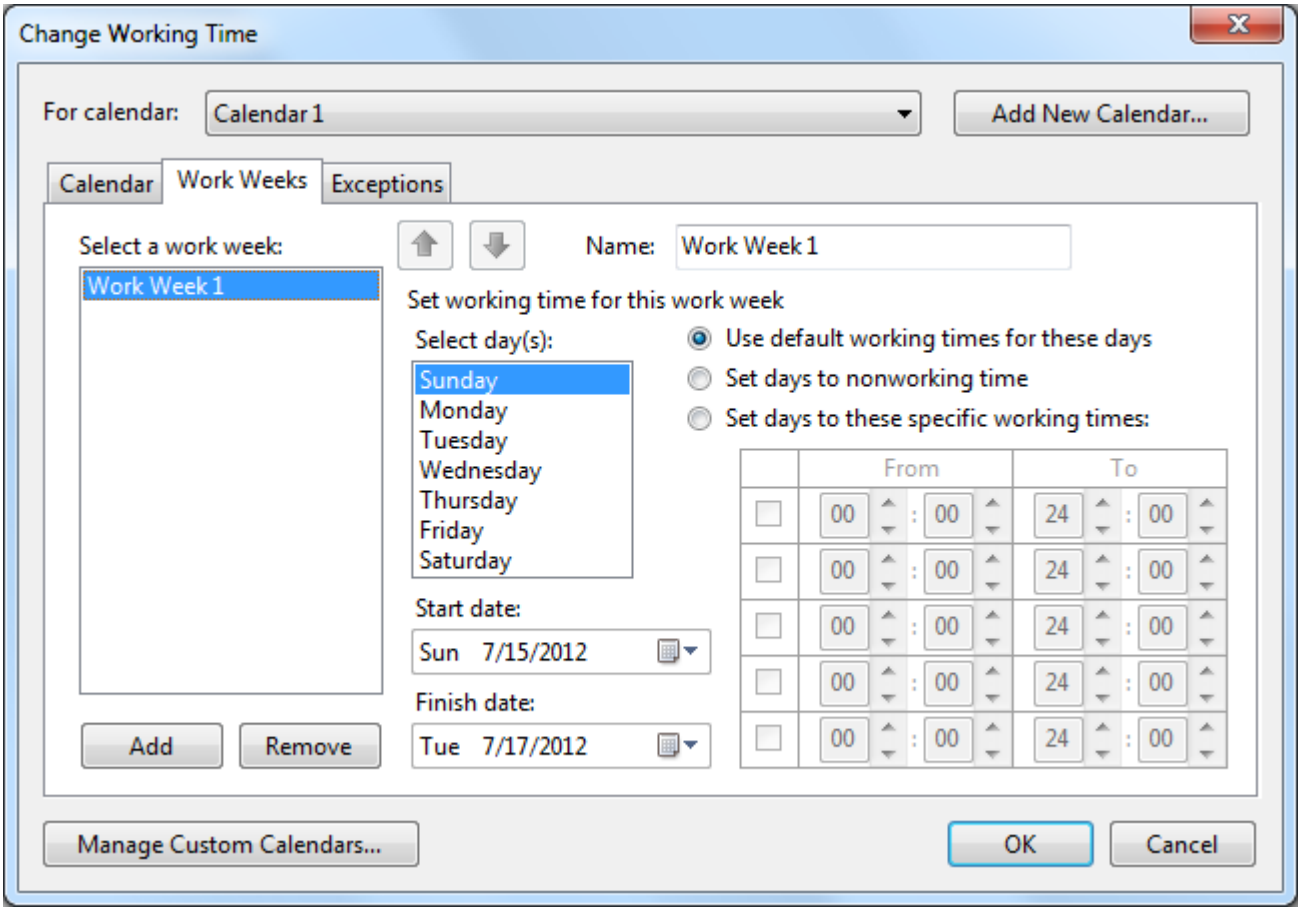
Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

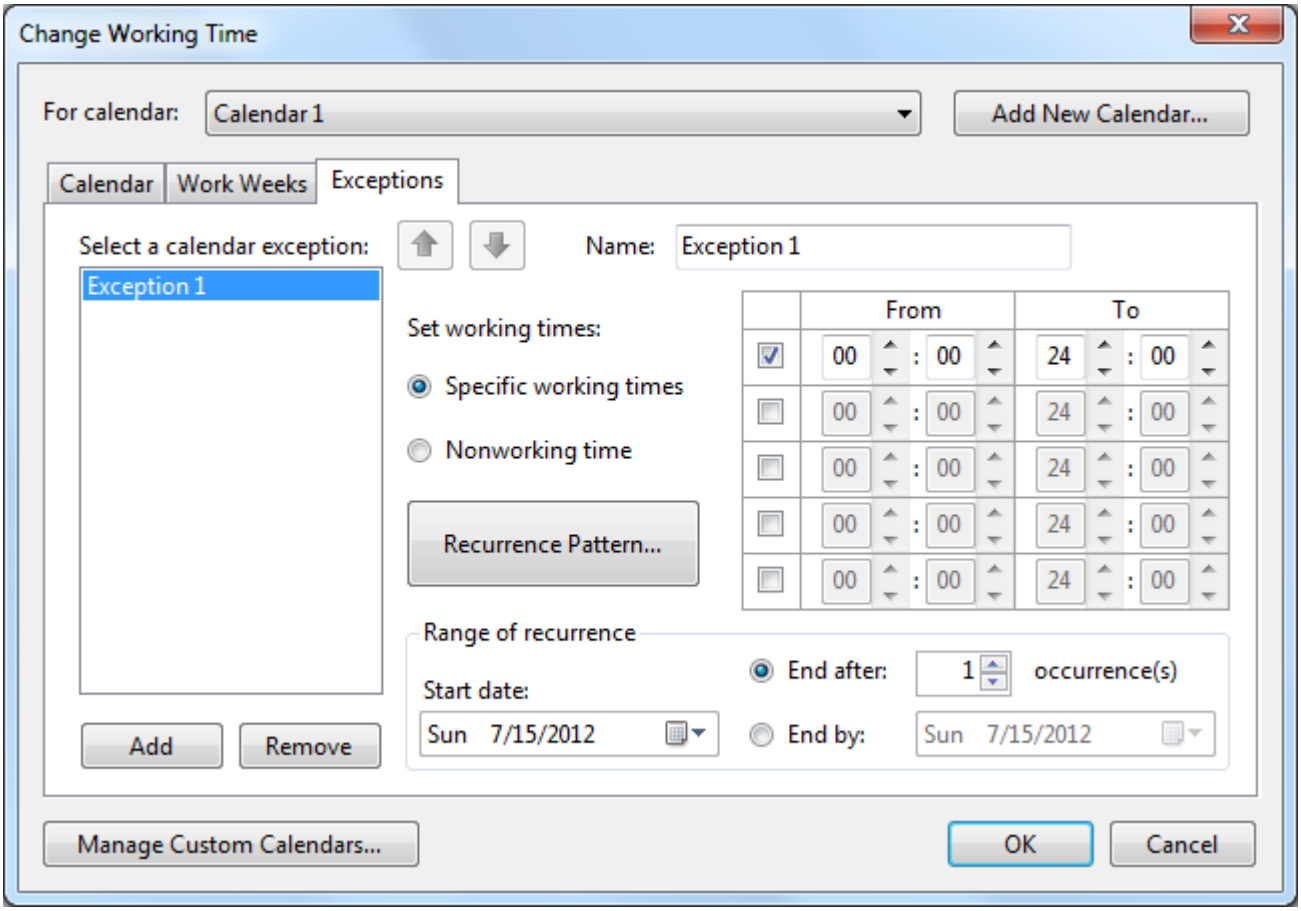
Once you select the day(s) you wish to change from a working day to non-working day, or vice versa, you can choose one of the following:

- 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.

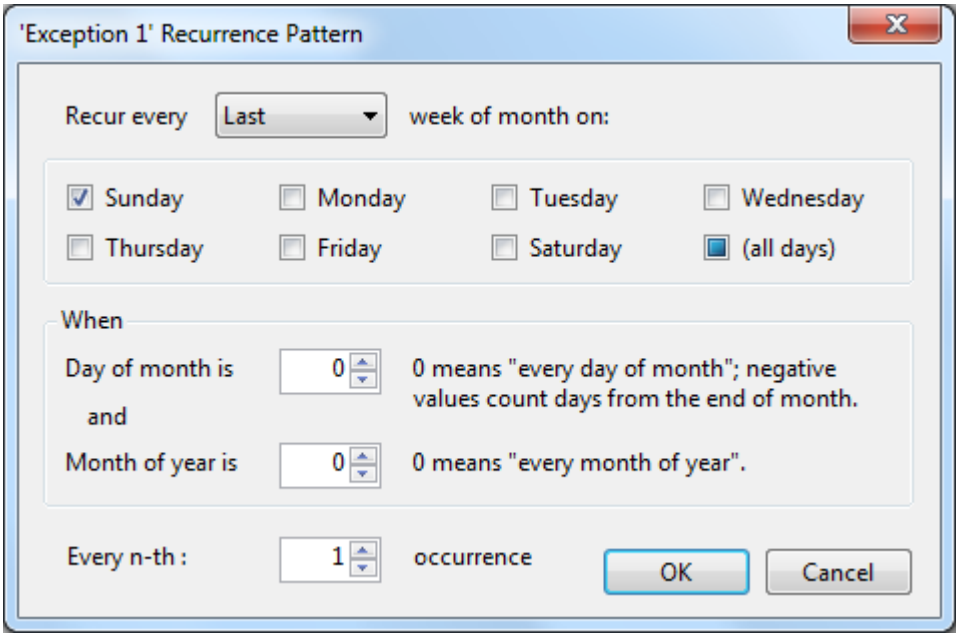
Exceptions Tab

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

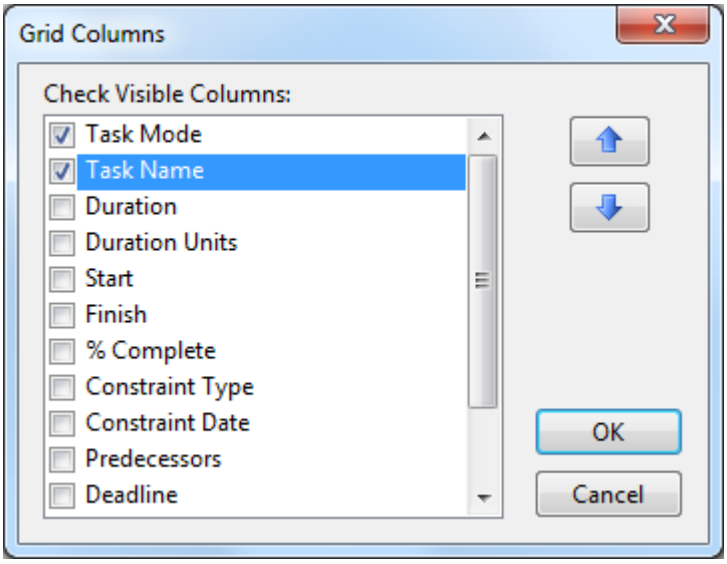
Any tasks scheduled around the calendar exception will automatically be rescheduled to take into account the nonworking time of the exception.

Grid Columns Dialog Box

The **Grid Columns dialog box** is used for specifying which attribute names appear in the columns of the **C1GanttView** 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 C1GanttView Toolbar.



Page Setup Dialog Box

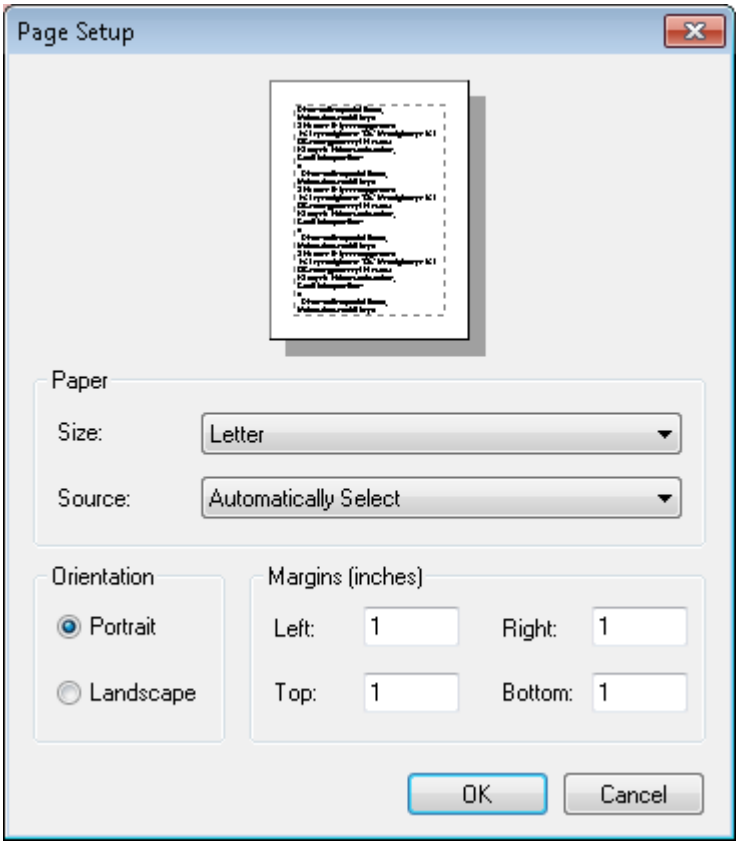
The **Page Setup** dialog is used for selecting the paper size and paper source to be used as well as the orientation and margin.

Access the Page Setup dialog

Click on the **Print** button from the Ganntview toolbar and then click on the **Page Setup** button in the Print Dialog box.

The **Page Setup** dialog box appears like the following:

Copyright © 2017 GrapeCity, inc. All rights reserved.



The **Page Setup** dialog box consists of the following **Paper** groupbox, **Orientation** groupbox, and **Margins** groupbox. Click **OK** to apply the settings and close the **Page Setup** dialog box. If you want to cancel the settings then click the **Cancel** button.

Paper Size

The size of the paper to be used. Click the dropdown arrow to select the paper size you want.

Source

The source is the paper tray that will be used to feed the printer. By default, Automatically Select is chosen. This is the 8.5 x 11 paper size.

Orientation

Determines whether the paper will be displayed in Portrait orientation which is vertical or Landscape which is horizontal. Portrait is the default setting.

Margins

Extra space on the edge of every printed page. The default setting is 1 inch for the **Left**, **Right**, **Top**, and **Bottom** margins. If you need to print more information on the page reduce the margins or if you need less information on the page increase the margins.

Print Preview Dialog Box

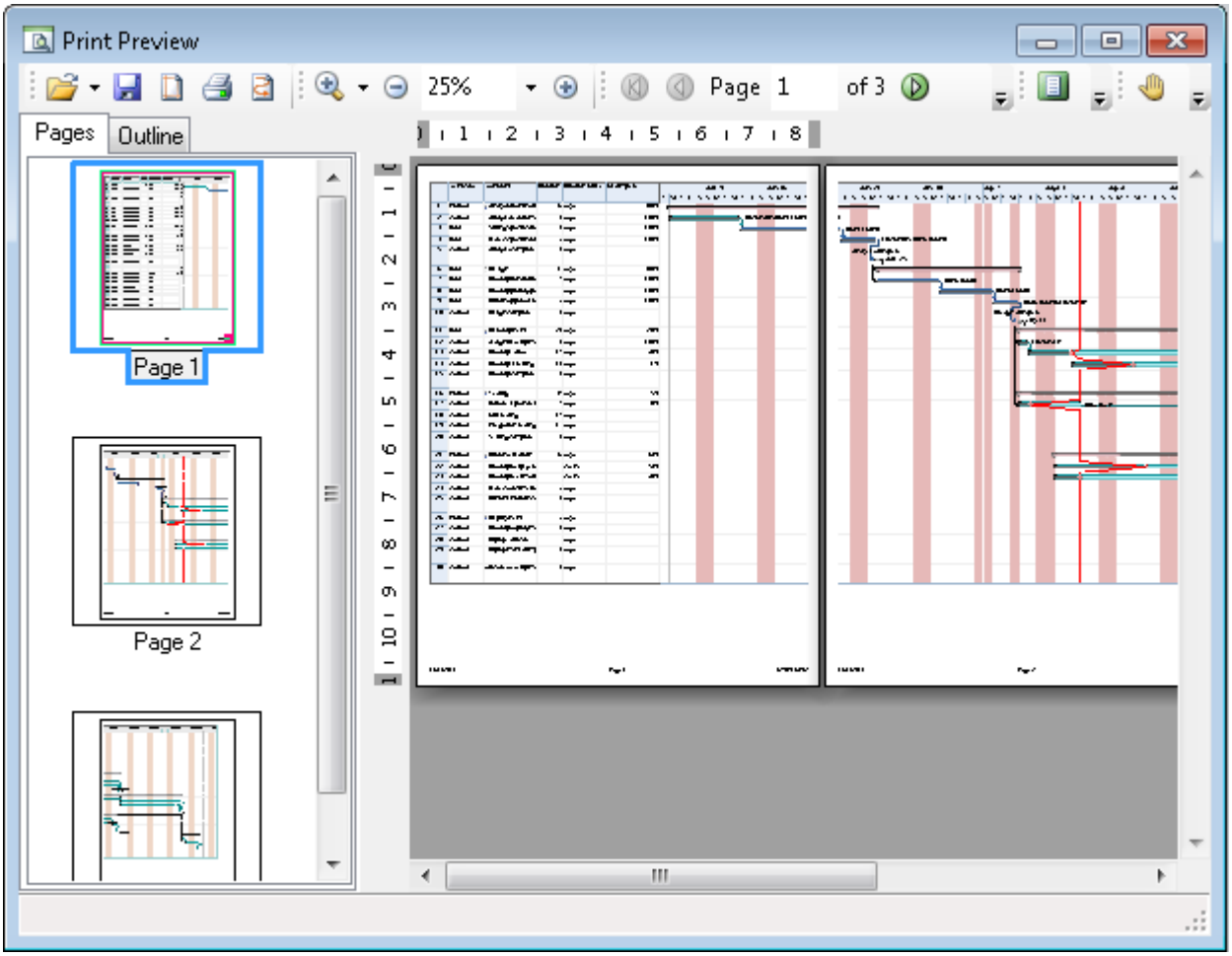
The **Print Preview** dialog box is used for customizing the header, footer, and legend appearance in each page.

Access the Print Preview dialog box

Click on the **Print** button in the GanttView toolbar and then click the Preview button from the Print dialog.

The **Print Preview** dialog box appears like the following:

Copyright © 2017 GrapeCity, inc. All rights reserved.



The Print Preview dialog includes the following components:

- Open File button - Click the Open File button to open a different .c1d or .c1dx document to preview.
- Save button - Click the Save button to save button to save the current .c1d or .c1dx document.
- Page Setup button - Opens the Page Setup dialog where you can change the page settings.
- Print button - Prints the document.
- Reflow button - Refreshes the document.
- Zoom-in/Zoom-out button - You can select the Zoom-in or Zoom-out button.
- Zoom-out button - Zooms out the current page in the Print preview dialog.
- Zoom Percent - Displays the current zoom-out or zoom-in percent.
- Zoom-in button - Zooms in the current page in the Print preview dialog.
- Go to first page button - Navigates to the first page.
- Go to previous page button - Navigates to the previous page.
- Page number textbox Page 2 of 3 - Displays the current page number of the total pages.
- Pages tab - This tab includes thumbnails of all the pages in the document.
- Outline tab - This tab includes an outline view of all the pages in the document.

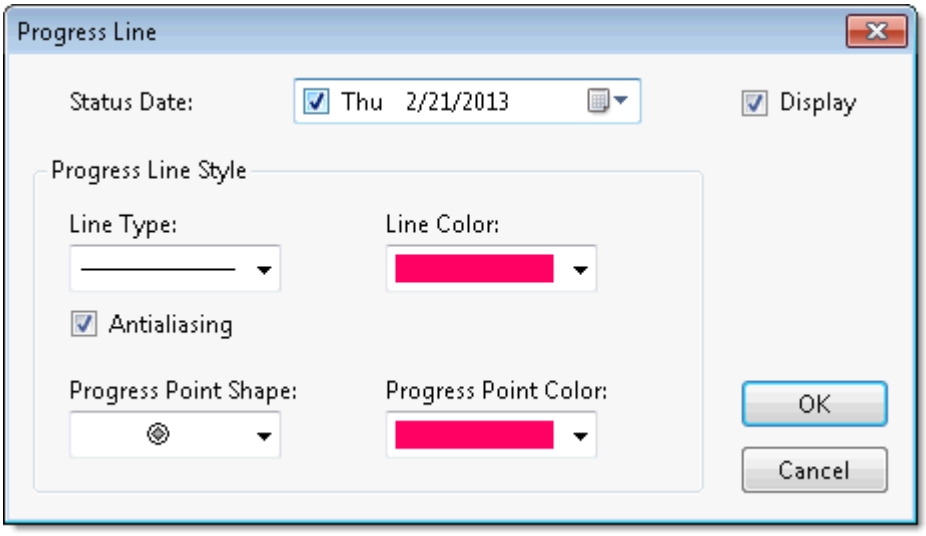
Progress Line Dialog Box

Copyright © 2017 GrapeCity, inc. All rights reserved.

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

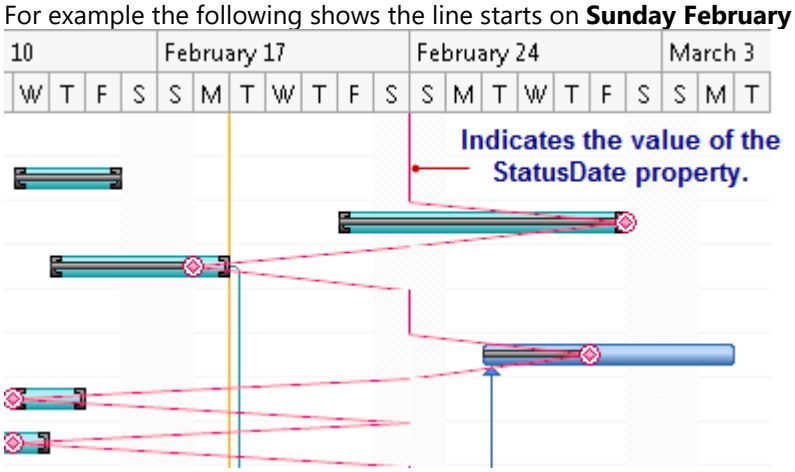
To Access the Progress Line dialog box

Click on the **Progress Line** button,  , in the **C1GanttView** Toolbar.



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.

Print Dialog Box

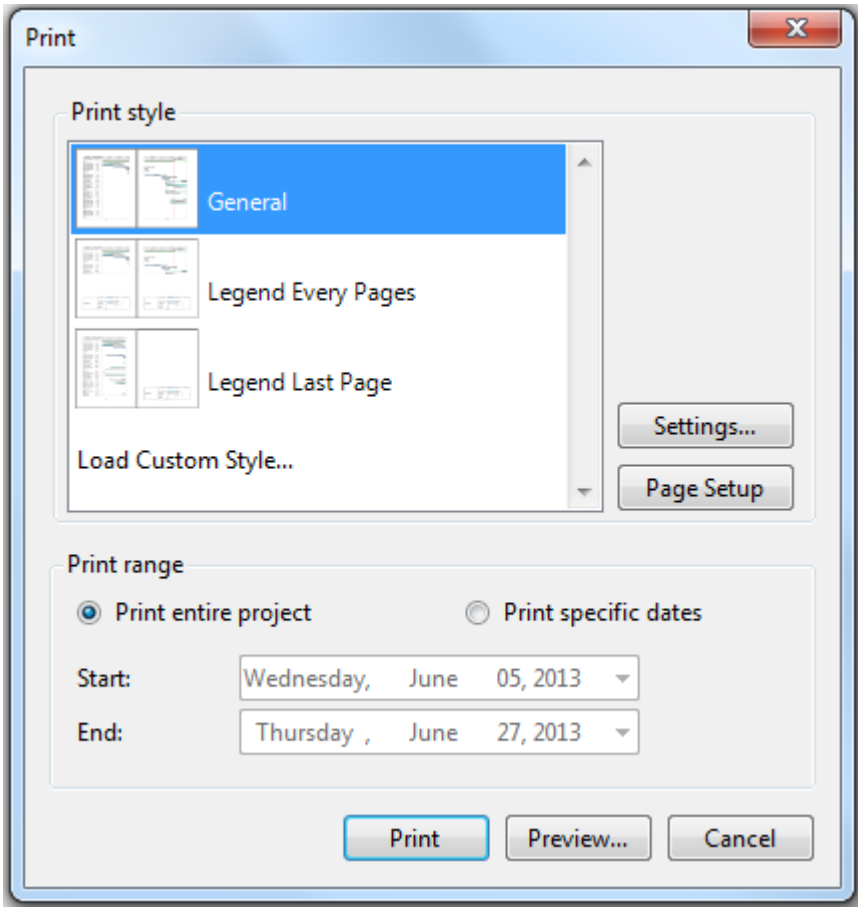
The **Print** dialog box is used for managing the printing such as selecting the print style, specifying the content and font settings for the Header, Footer, and/or Legend, specifying page setup, and determining print range.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Access the Print dialog box

Click on the **Print** button,  , in the **C1GanttView** Toolbar.

The **Print** dialog box appears like the following:



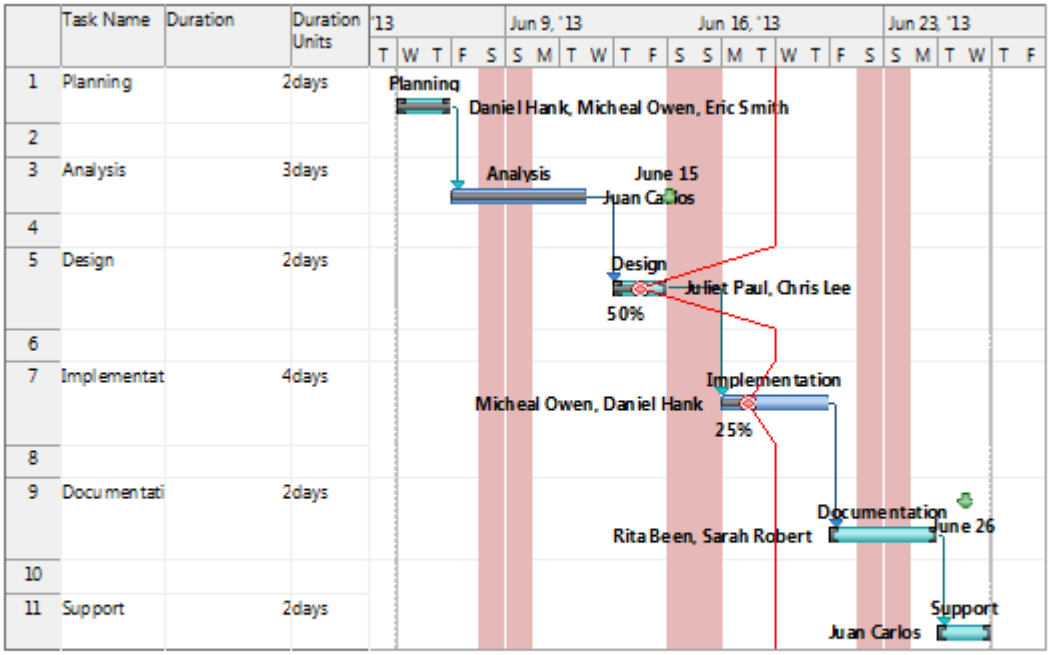
The Print dialog box consists of the **Print style** groupbox, **Print range** groupbox, **Print** button, **Preview** button, and **Cancel** button. Each consituent operates as follows:

The Print Style groupbox

In the Print style groupbox you can select the type of print style to determine your page layout. There are 4 different print styles to choose from: General, Legend Every Pages, Legend Last Page, and Load Custom Style...

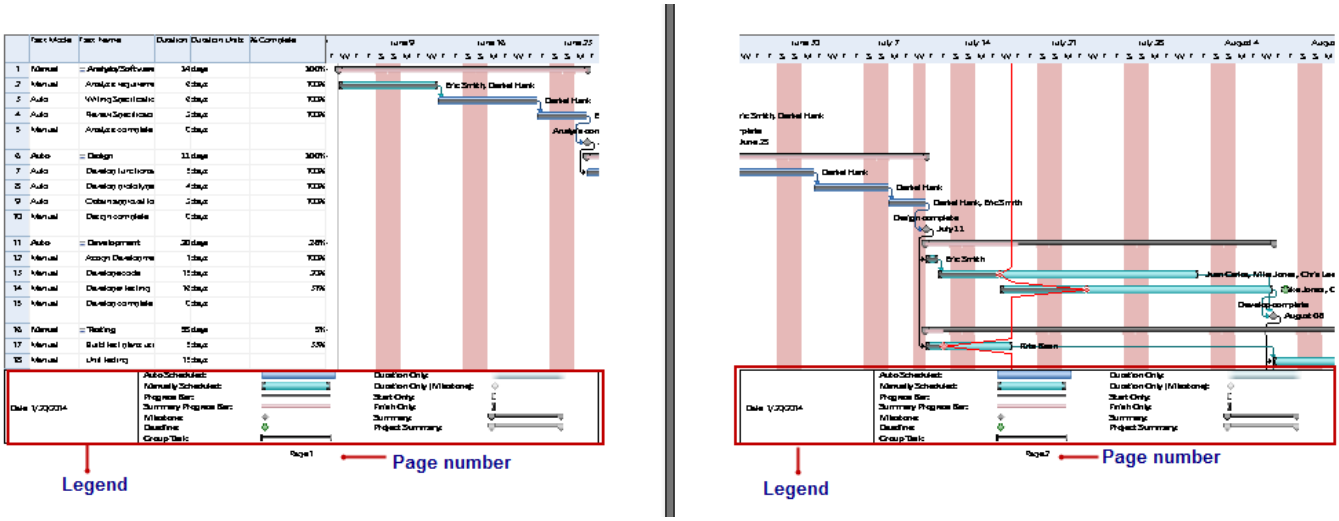
The **General** layout provides the layout that only includes GanttChart. This is the default style. The **General** layout appears like the following:

Copyright © 2017 GrapeCity, inc. All rights reserved.



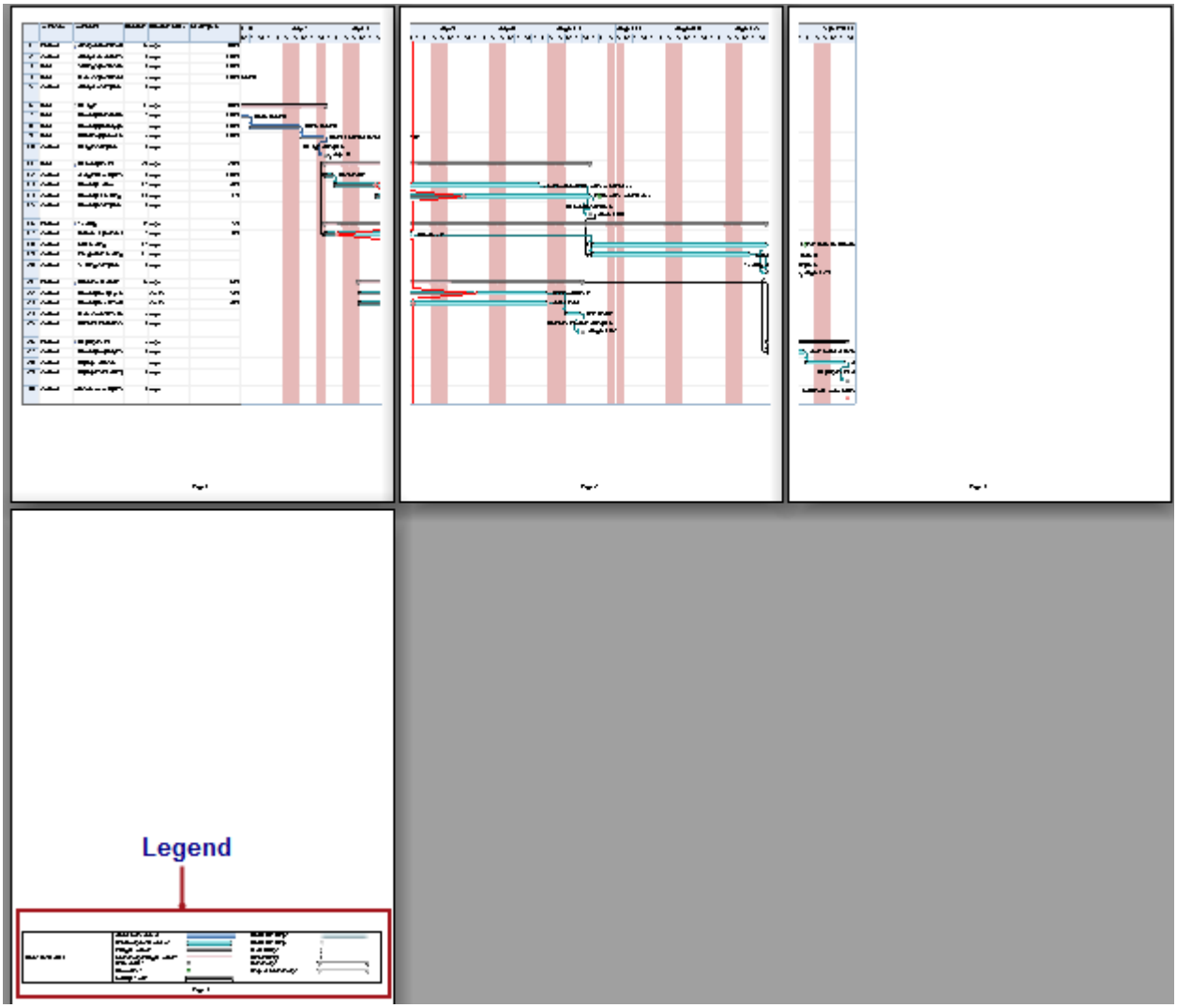
The page number field appears in the Footer center, by default.

The **Legend Every Page** layout includes the GanttChart and Legend on the last page. The Legend Every Page appears like the following:



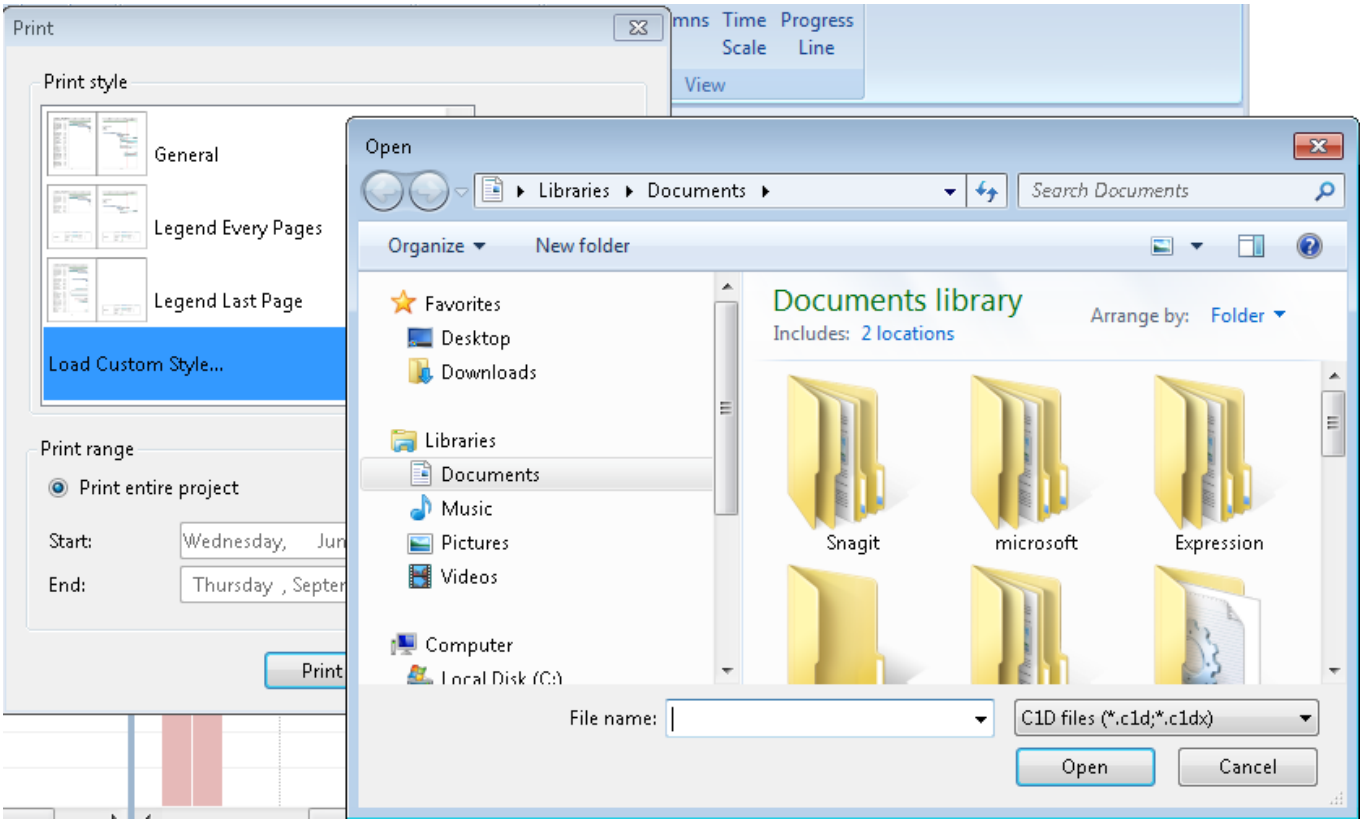
The Legend Last Page layout include the GanttChart on every page and the Legend on the last page. The **Legend Last Page** layout appears like the following:

Copyright © 2017 GrapeCity, inc. All rights reserved.



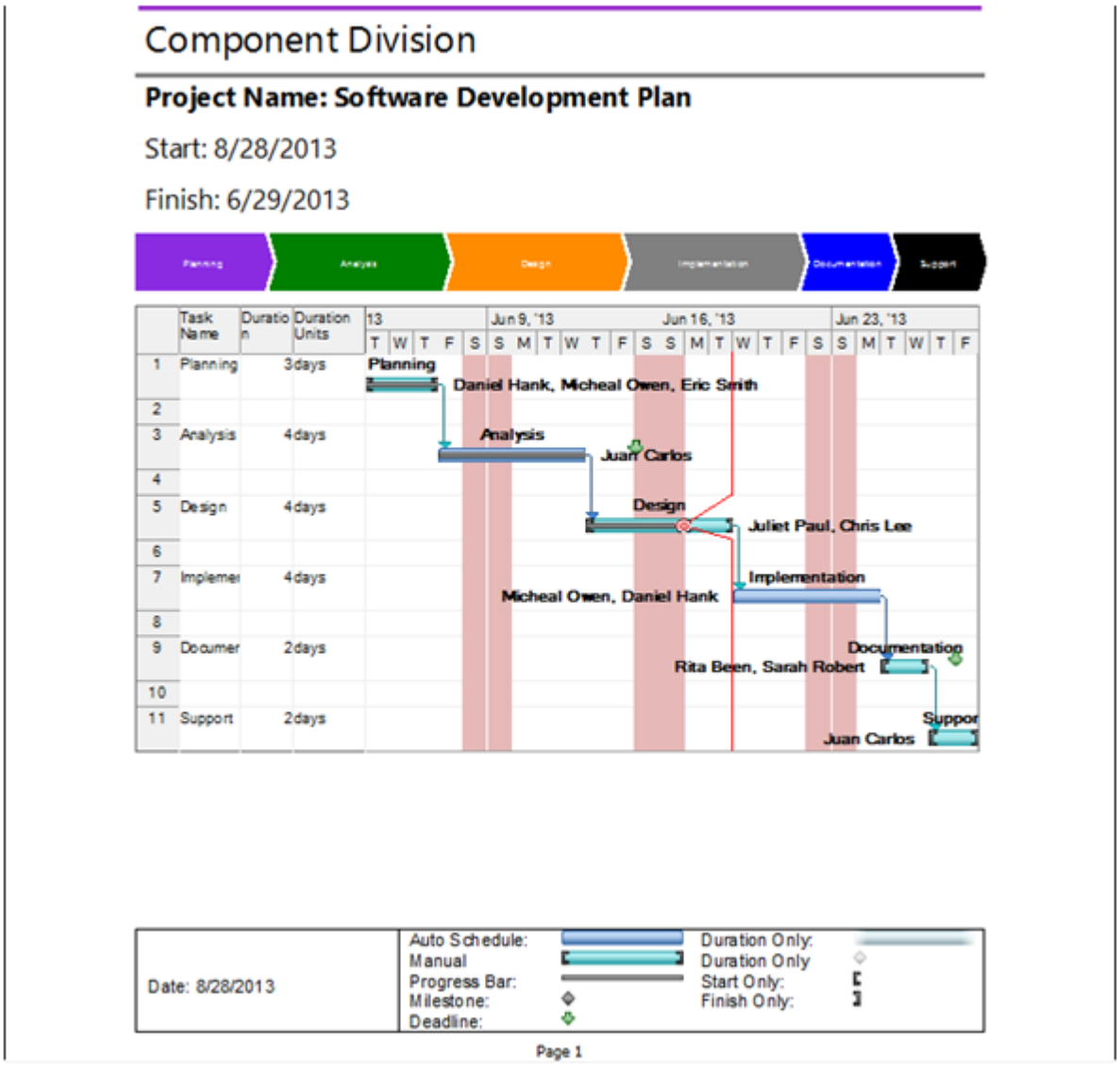
The Load Custom Style opens the **Open File** dialog where you can load the custom style from the .c1d or .c1dx file.

Copyright © 2017 GrapeCity, inc. All rights reserved.



The following image illustrates the appearance of a custom print style:

Copyright © 2017 GrapeCity, inc. All rights reserved.



Once the layout is selected you can click on the **Settings** button to invoke the Style Settings dialog where you can customize the header, footer, and legend appearance.

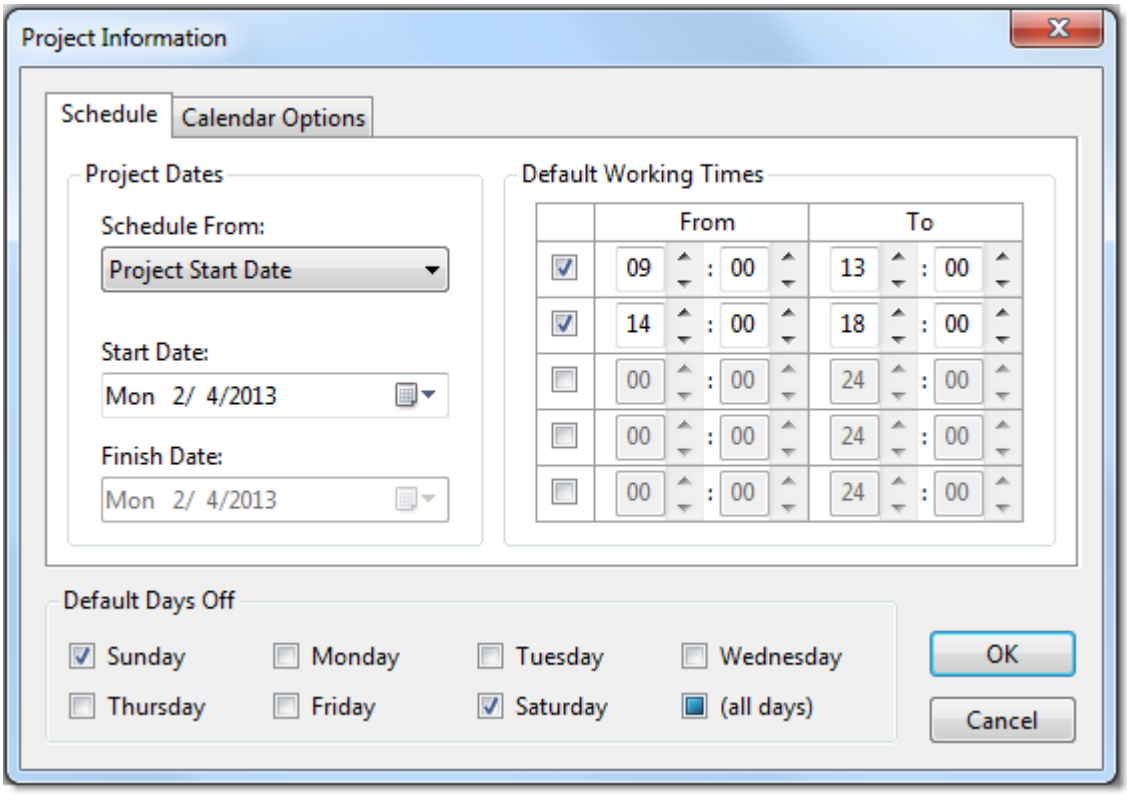
Project Information Dialog Box

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 **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.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

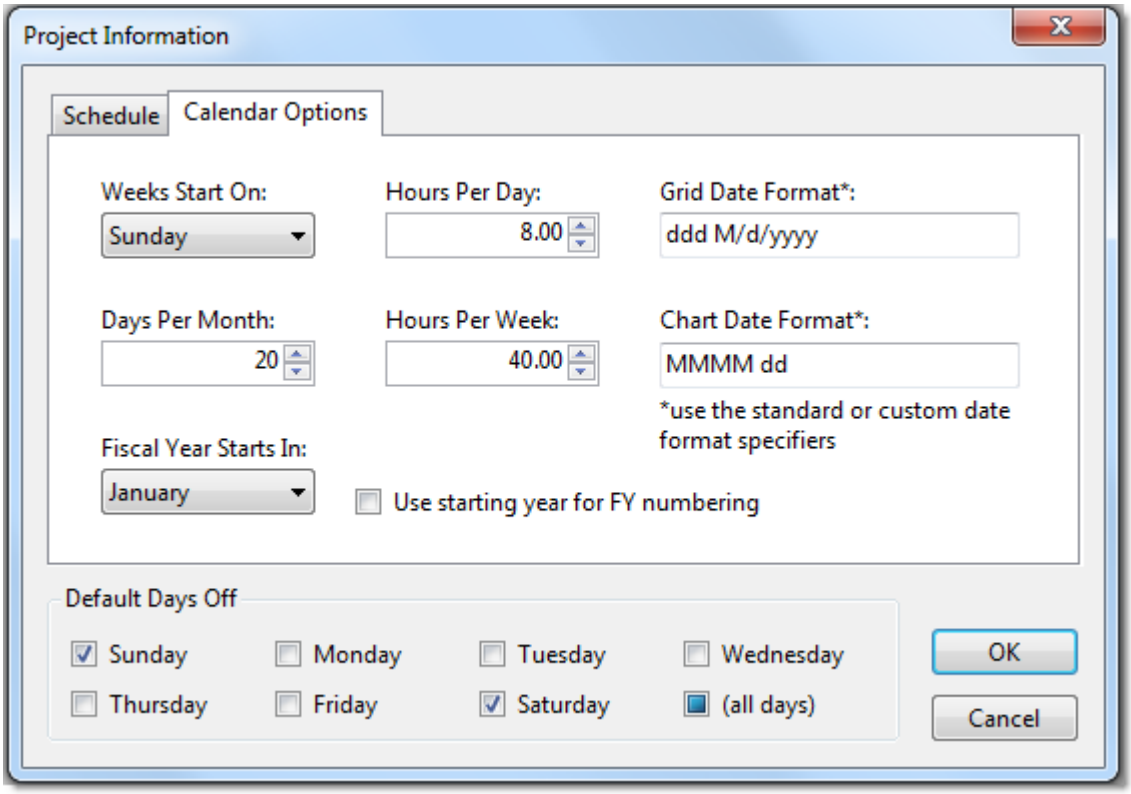
  > **Note:** If you need to find out how late you can start a project set the Schedule From to **Project Finish Date** and then once you begin the project set it back to **Project Start Date**.

- **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:

Copyright © 2017 GrapeCity, inc. All rights reserved.



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

Copyright © 2017 GrapeCity, inc. All rights reserved.

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.

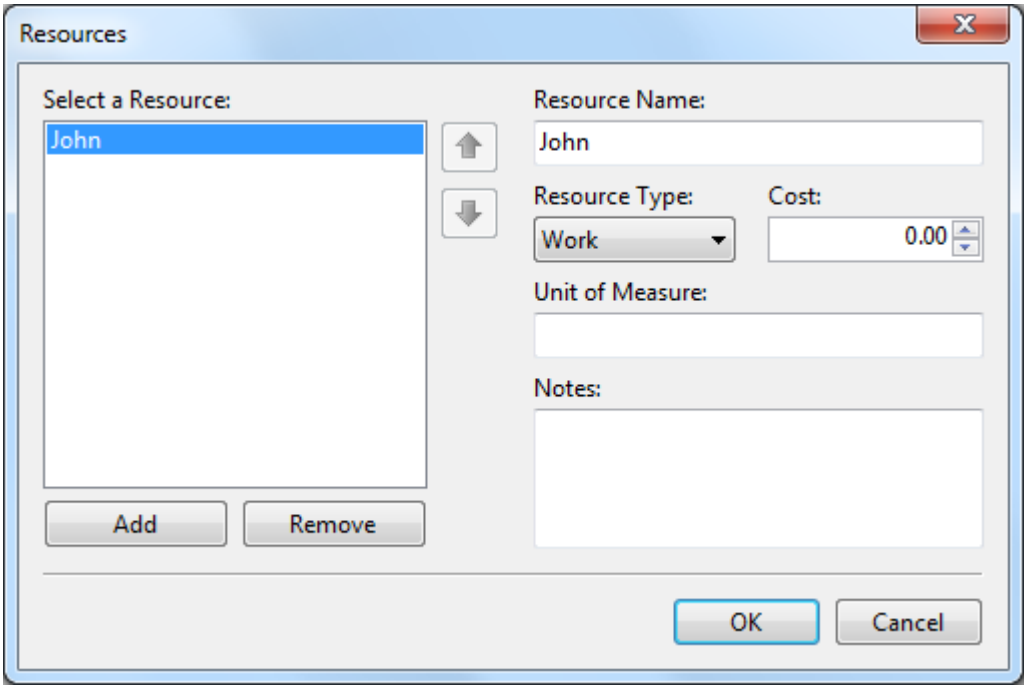
Cost

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.



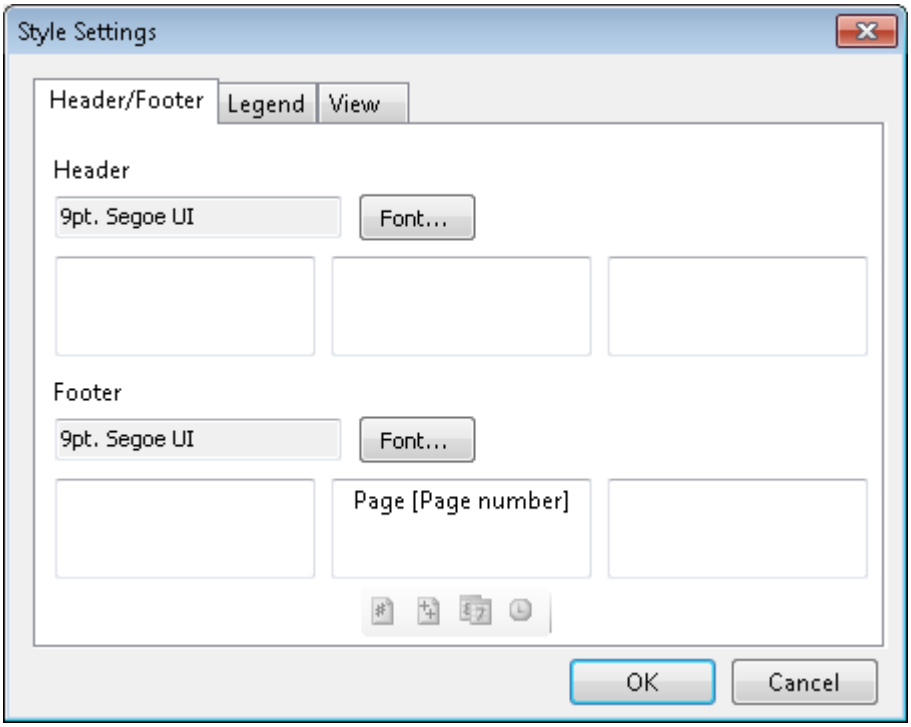
Style Settings Dialog Box

The **Style Settings** dialog box is used for customizing the header, footer, and legend appearance in each page.

Access the Style Settings dialog box

Click on the **Settings** button in the Print Dialog box.

The **Style Settings** dialog box appears like the following:



The Style Settings dialog box consists of the following tabs:

Copyright © 2017 GrapeCity, inc. All rights reserved.

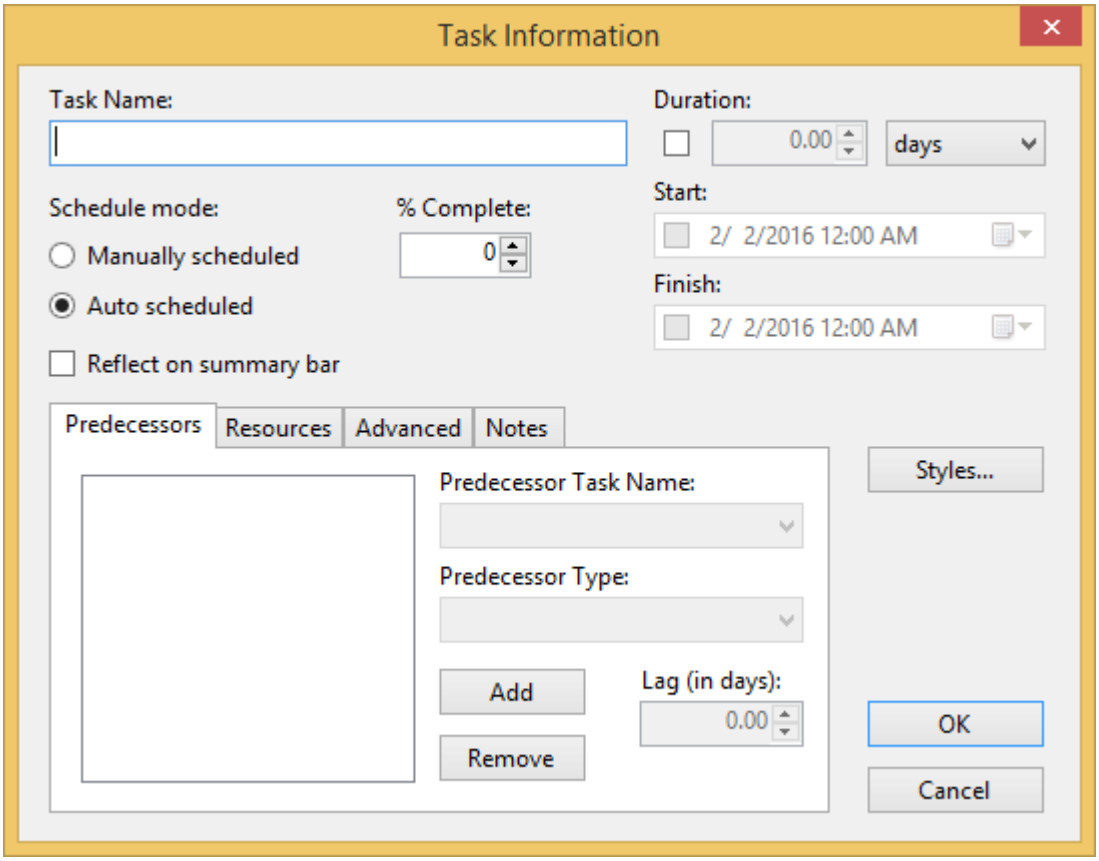
- **Header/Footer** - This tab is used to edit the Header and Footer. You can select font/color for the Header or Footer's text. There are three text boxes to input text content for left/center/right part. When editing content of each part, you can click to four built-in buttons to insert page number, page count, date, time field, respectively.
- **Legend** - This tab is used to edit the text area of Legend, this area is also divided into three parts: left/right/center... and they're customize the same ways that have done with the header/footer. The text area's width is adjusted by changing the value in the numeric up down box with inch unit. For the label area, user can only change the font/color of text by clicking to Legend Label button
- **View** - This tab you can select the number of GridView's columns to be printed and the number of repeat columns among them. By default, all visible columns will be printed.

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.



See Inserting a Task for an example.

Predecessors Tab

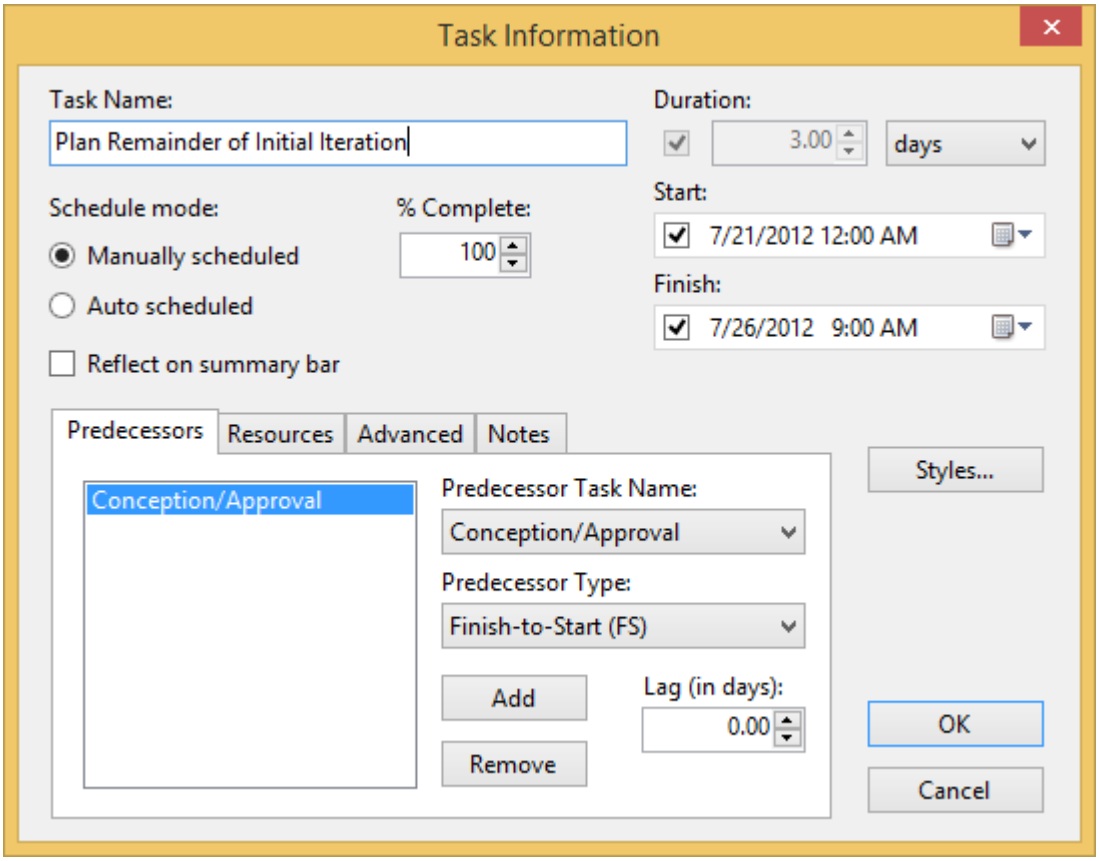
In the Predecessors tab you can add or remove predecessors, select the predecessor task name, select the

Copyright © 2017 GrapeCity, inc. All rights reserved.

predecessor type, and specify the lag time in days.

For conceptual information on predecessors see, Task Predecessor.

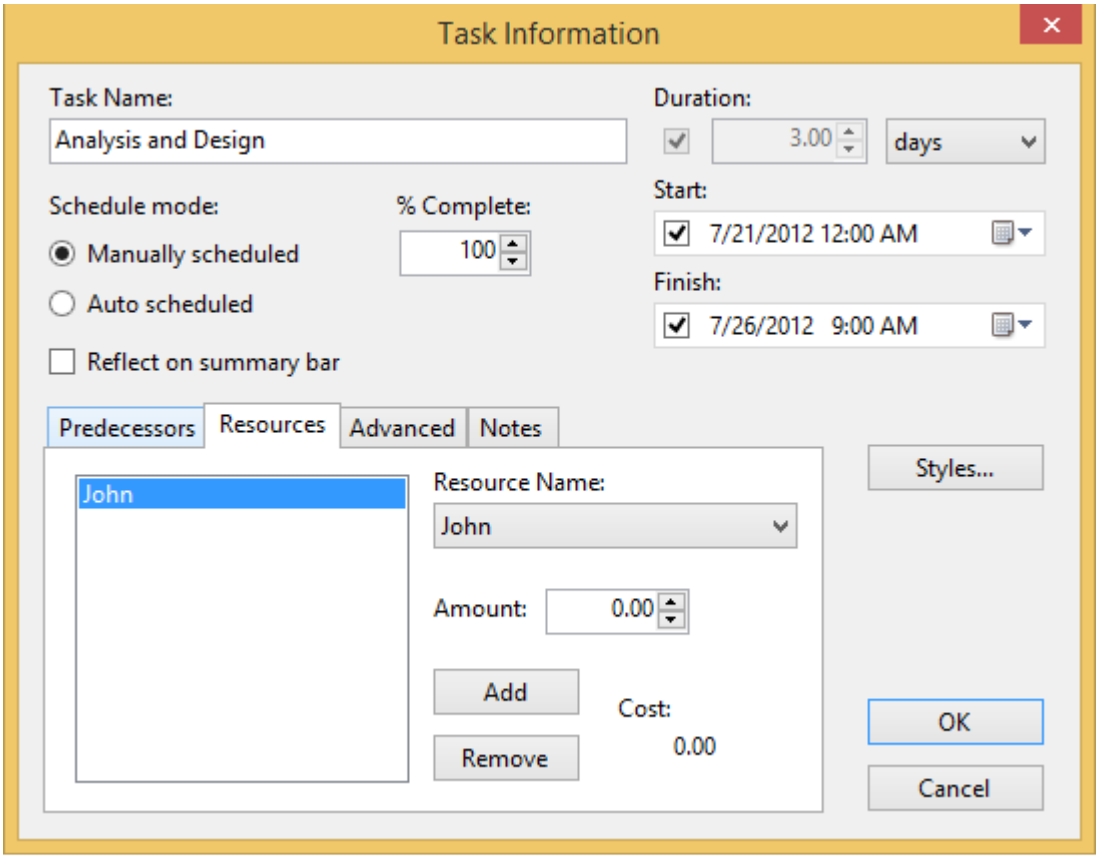
For procedural information on predecessors see, Creating Predecessors.



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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

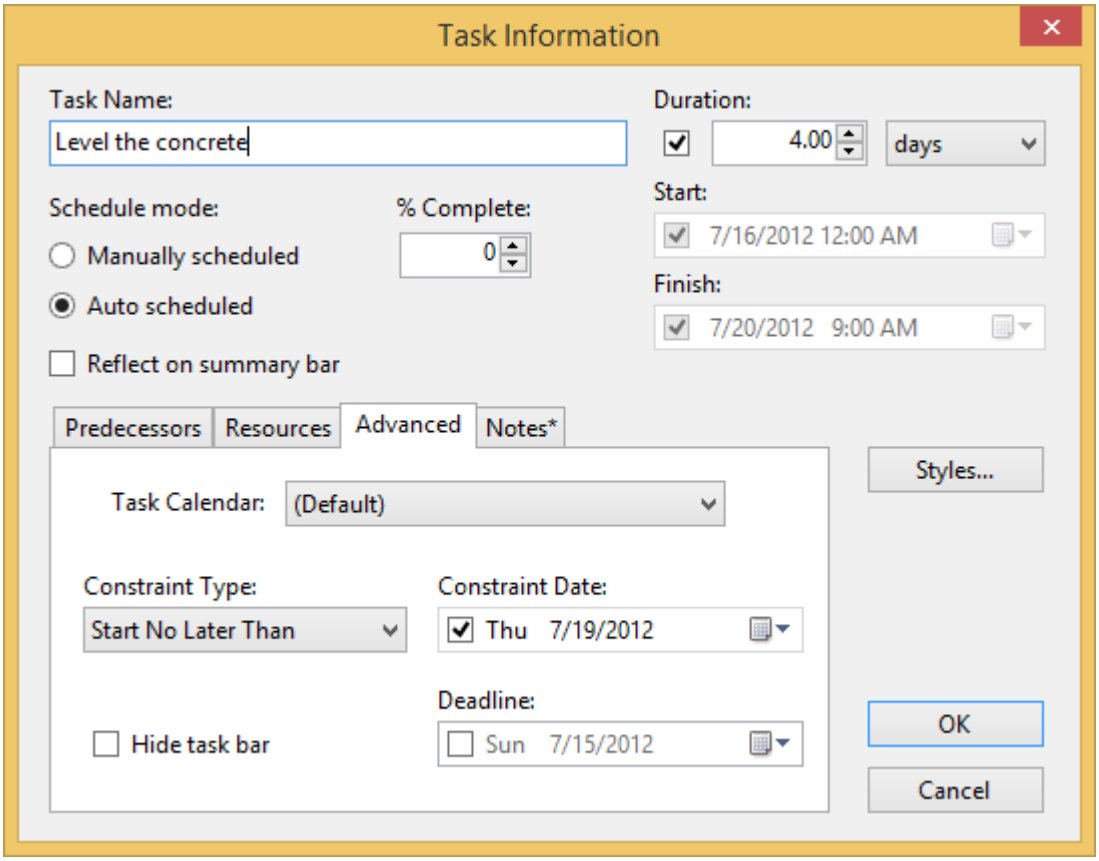


Advanced Tab

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.

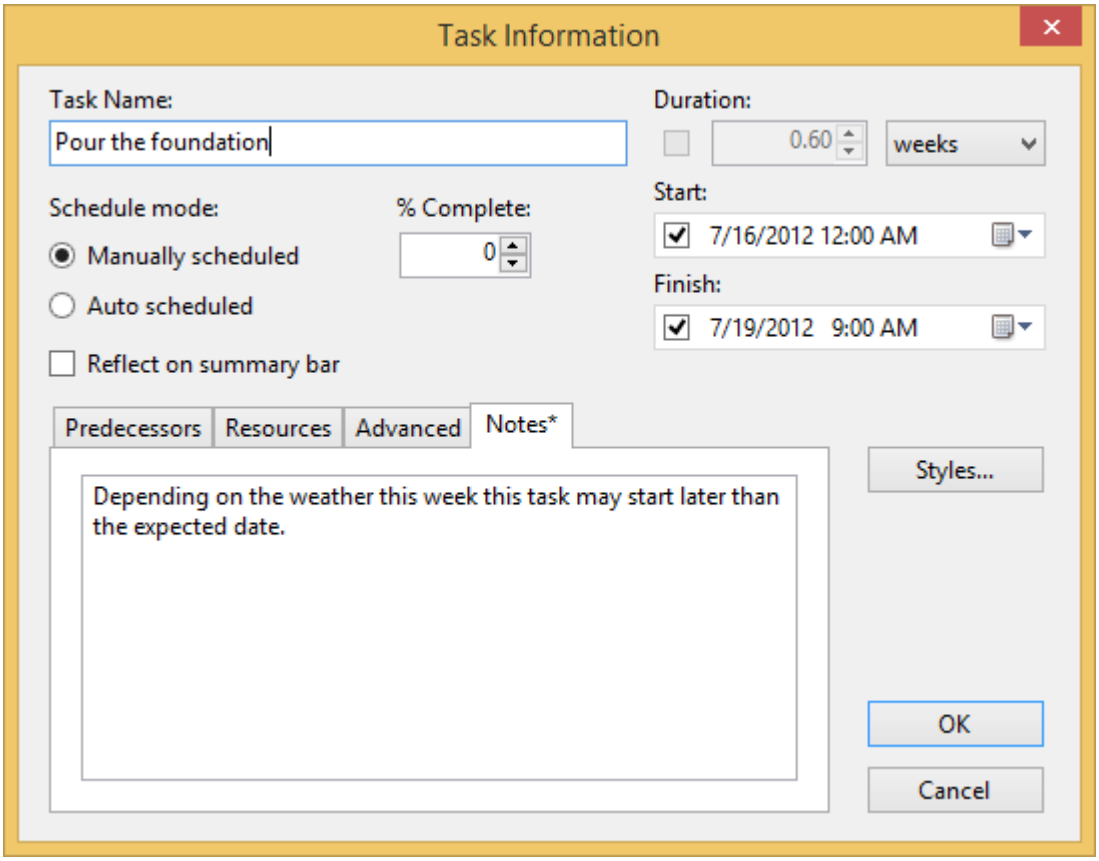
Copyright © 2017 GrapeCity, inc. All rights reserved.



Notes Tab

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

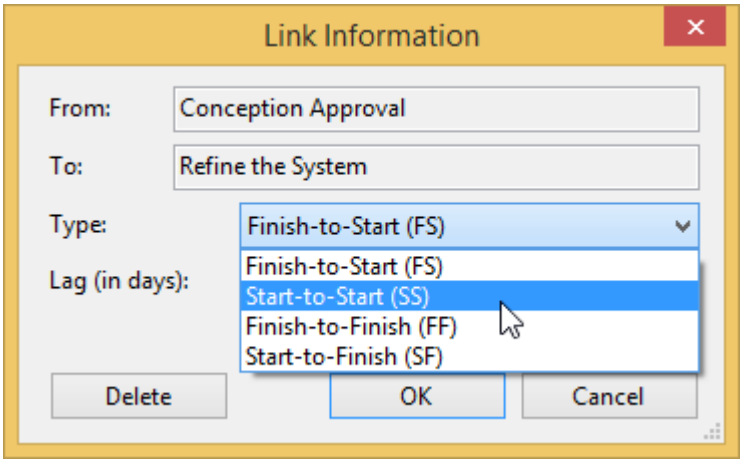
Copyright © 2017 GrapeCity, inc. All rights reserved.



Link Information Dialog Box

The **Link Information** dialog box displays the information about predecessors.

In the Link Information dialog box, you can edit or delete the predecessors information such as the **From** and **To** tasks, **Type** of predecessor, and **Lag** time in days. The Link Information dialog box can be assessed by double-clicking the predecessors line.



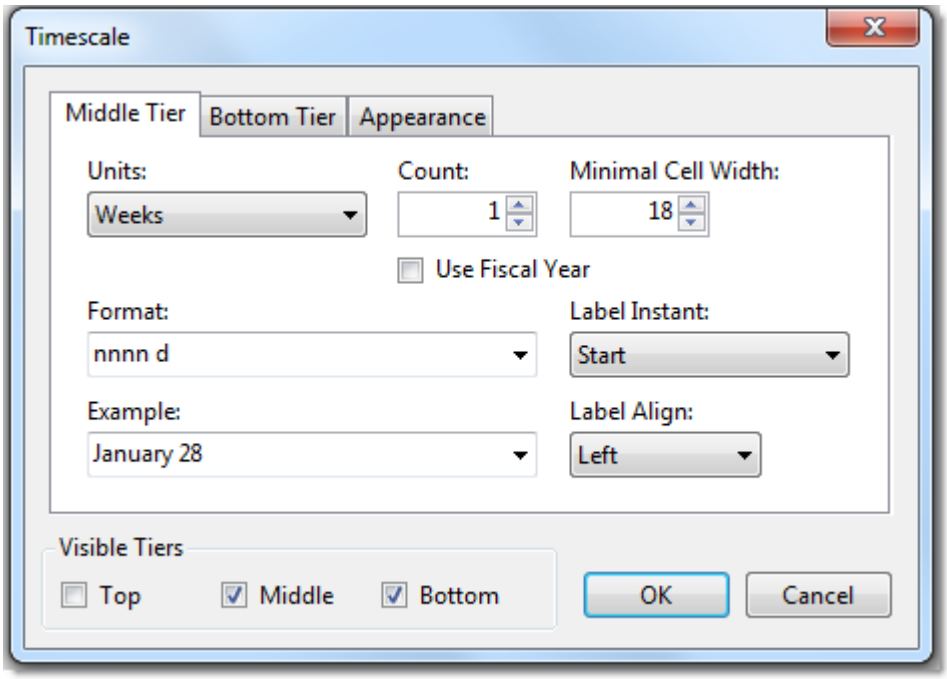
Timescale Dialog Box

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

To Access the Timescale dialog box

Click on the **Timescale** button in the C1GanttView Toolbar.



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 for an example.

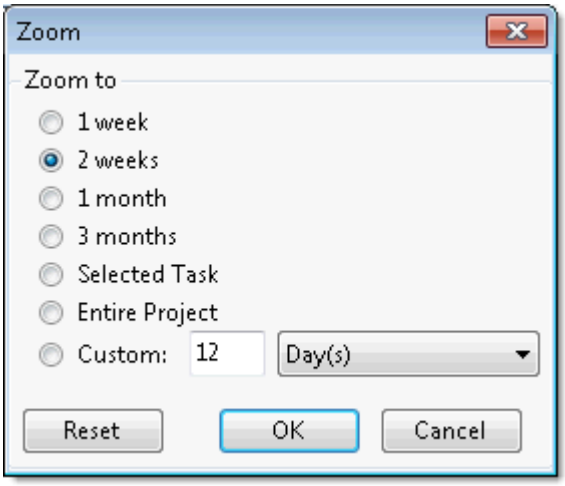
Zoom Dialog Box

The **Zoom** dialog box is used for changing the scale view of tiers on the chart view.

To Access the Zoom Time dialog box

Click on the **Zoom** button,  , in the C1GanttView Toolbar.

Copyright © 2017 GrapeCity, inc. All rights reserved.

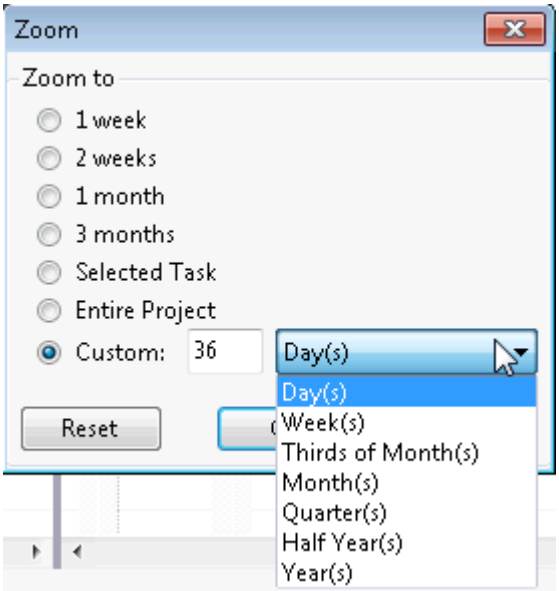


You can specify the time range you wish to zoom to from selecting one of the following options or you can specify your own time range in the custom option:

- 1 week
- 2 weeks
- 1 month
- 3 months
- Selected Task
- Entire Project
- Custom

The zoom operations will adjust the timescale and start date of Chart View to display the entire time range in the visible area.

Besides the predefined time range, users can zoom to a specific time range by selecting the **Custom** option in the Zoom dialog box, then enter a custom time range such as 36 in the **Count** and select **Unit** (Day(s),Week(s), Thirds of Month(s), Month(s), Quarter(s), Half Year(s), Year(s). The maximum value of **Count** in the input text box is **9999**.



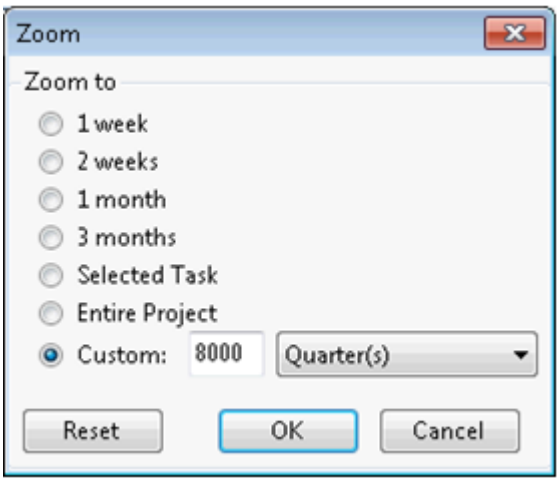
When Zoom dialog is loading, the number in the text box will have the initial value based on the current duration that the chart view is representing.

- The Reset button resets the value in the custom text box to the initial value.
- The number in the text box must satisfy the following conditions:
- Is positive numbers.

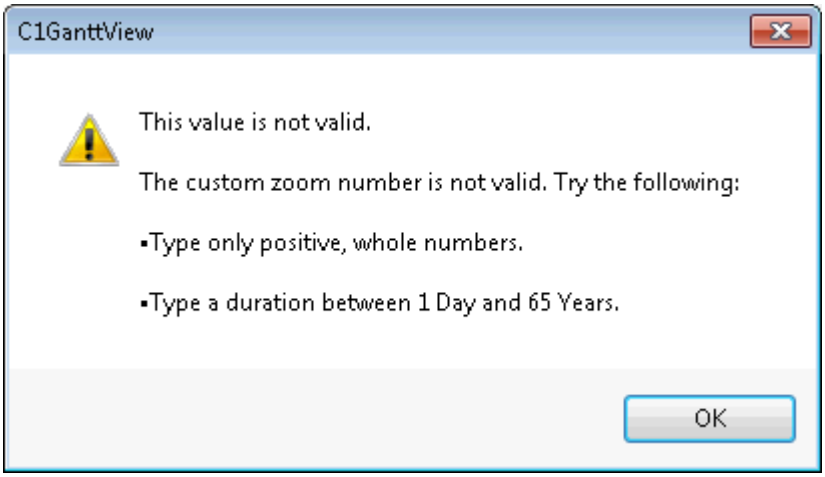
Copyright © 2017 GrapeCity, inc. All rights reserved.

- The zoom duration must between 1 Day and 65 Years.

For example if you specify an integer in the Count textbox outside the valid range, such as the following:



The following error message will be shown:



The following table displays the default maximum count for each unit used:

Unit	Max Count
Minutes	50
Hours	24
Days	7
Weeks	3
Third of Months	3
Month	3
Quarters	2
Half-Years	1
Years	50

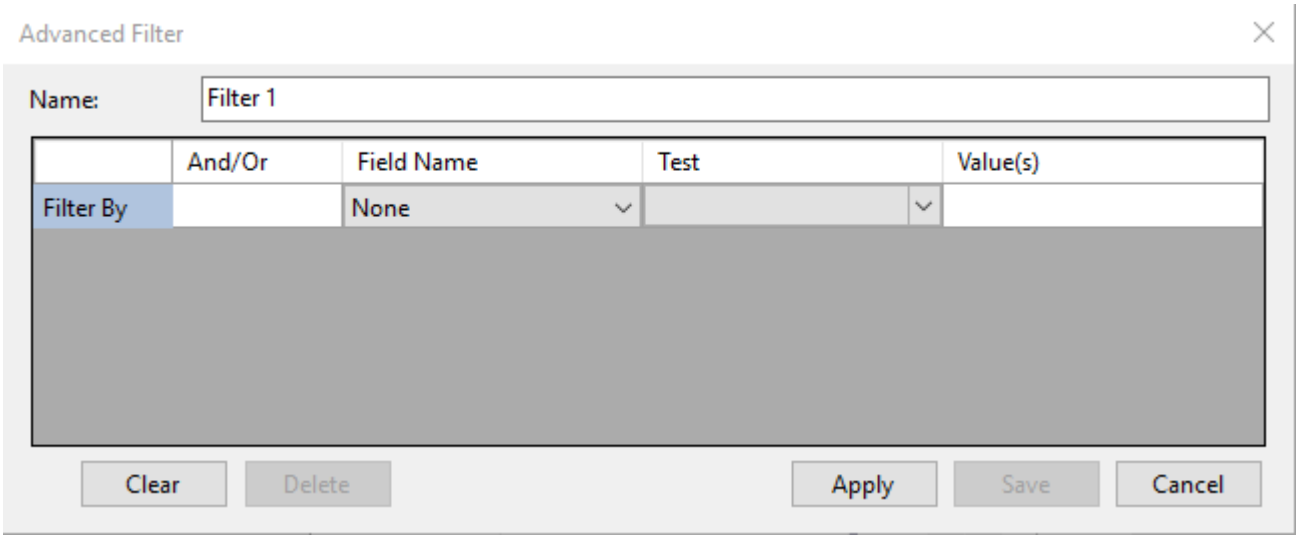
Copyright © 2017 GrapeCity, inc. All rights reserved.

If the current unit is the same as the selected unit that has been chosen in the **Zoom** dialog box then the **MaxCount**=50.

Advanced Filter Dialog Box

The **Advanced Filter** dialog is designed to create and apply custom filters in GanttView. The dialog appears on selecting the **Advanced Filter** option available under the **Filter** button on the toolbar.

The following image shows the Advanced Filter dialog:



The Advanced Filter dialog provides various options as follows:

Option	Purpose
And/Or	Select 'And' or 'Or' conditional operator from the drop-down list.
FieldName	Select the field for filtering from the drop-down list.
Test	Select the test condition from the drop-down list.
Value(s)	Displays the values corresponding to the field selected in the FieldName option. For example, if the FieldName is set to Mode, the Value(s) drop-down menu shows Auto and Manual options for selection.
Clear	Click the button to clear the filter.
Apply	Click the button to apply the filter.
Save	Click the button to save the filter.
Cancel	Click the button to close the dialog.
Delete	Click the button to delete filter line.

C1GanttView Toolbar

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

Copyright © 2017 GrapeCity, inc. All rights reserved.

The **C1GanttView** 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.
	Undo	Undo the last action performed.
	Redo	Redo the last action performed.
	Grid Columns	Opens the Grid Columns dialog box.
	Move task up	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.
	Field Styles	Opens the Field Styles dialog box.
	Add task	Opens the Task Information dialog box where the New Task is entered in the **Task Name:** textbox.
	Add New Summary Task	Add a new summary task to the gantt chart.
	Inactive	Inactivates the selected task in grid.
	Add blank row	Adds a blank row to task grid.
	Delete task	Deletes the selected task from the grid.
	Outdent Task	Increases the outline level of a task.
	Indent Task	Decreases the outline level of a task.
	Show Project Summary	Shows/Hides the project summary task.
	Group By	Opens the Group By menu where you can select from Task Mode, Task Complete vs. Task Incomplete, Duration, Milestones, Resources, Status, and Constraint Type. You can also clear groups, add new group by, or maintain hierarchy.

Copyright © 2017 GrapeCity, inc. All rights reserved.

	Filter	Opens the Filter drop down.
	Sort	Opens the sort drop down that provides options to sort the list by **Name**, **Start Date**, **Finish Date**, **Duration**, and **Remove Sort** and **Sort By** options.
	Project information	Opens the Project Information dialog box.
	Change Working Time	Opens the Change Working Time dialog box.
	Progress Line	Clicking the **Progress Line** button opens the Progress Line dialog box.
	Project Resources	Opens the Project Resources dialog box.
	Timescale	Opens the Timescale dialog box.
	Bar styles	Opens the Bar Styles dialog box.
	Scroll to task	Scrolls to the task within the grid.
	Zoom	Click on the dropdown arrow to choose one of the following commands: Zoom Out, Zoom In, or Zoom. Clicking Zoom opens the Zoom dialog box.
	Zoom Entire Project	Determines the time range by **StartDate** and **FinishDate** of project. The **ZoomToRange** function is used to zoom this range
	Zoom Selected Task	Determines the time range by the earliest **StartDate** and latest **FinishDate** of the selected task(s). After the time range is determined then the ZoomToRange function is called to zoom in to the given time range.
	Print	Opens the Print dialog box where you can specify the print style, settings, and page setup. Once your settings are selected you can then click on the print button to print the gantt chart.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Timescale Formats

The following table details the available format specifiers for the timescale labels. For more information see the Timescale Dialog Box 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 DateTimeFormatInfo.GetAbbreviatedMonthName.

Copyright © 2017 GrapeCity, inc. All rights reserved.

nnnn	Use DateTimeFormatInfo.GetMonthName.
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 DateTimeFormatInfo.GetShortestDayName.
www	Use DateTimeFormatInfo.GetAbbreviatedDayName.
wwww	Use DateTimeFormatInfo.GetDayName.
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

Copyright © 2017 GrapeCity, inc. All rights reserved.

	designator.
tt	Represents the AM/PM designator.
t{N1,N2}	Custom AM/PM designator.
Other Specifiers	
:	Separates the components of a time (use DateTimeFormatInfo.TimeSeparator)
/	Separates the components of a date (use DateTimeFormatInfo.DateSeparator)
“	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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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

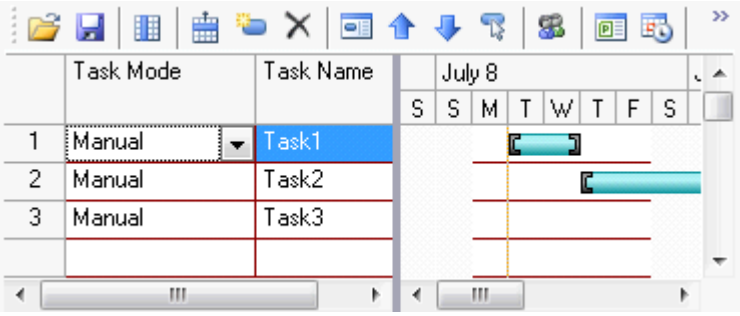
When the C1GanttView's **VisualStyle** property is set to **Custom** you can style any of its elements using the following properties:

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

CellBorderColor Example

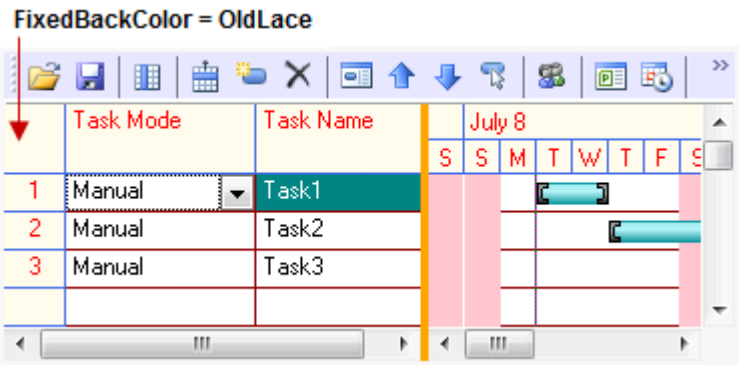
The following image illustrates the **CellBorderColor** property modified to DarkRed.

Copyright © 2017 GrapeCity, inc. All rights reserved.



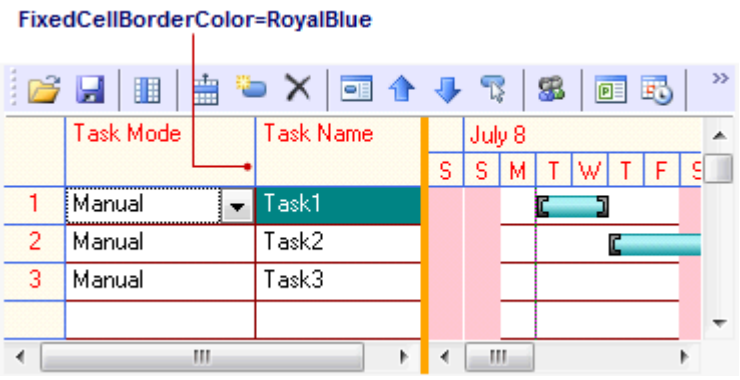
FixedBackColor Example

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



FixedCellBorderColor Example

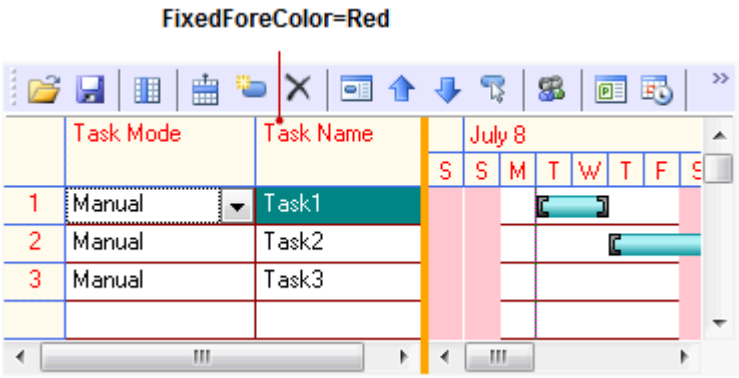
The following C1GanttView illustrates the **FixedCellBorderColor** modified to **RoyalBlue**.



FixedForeColor Example

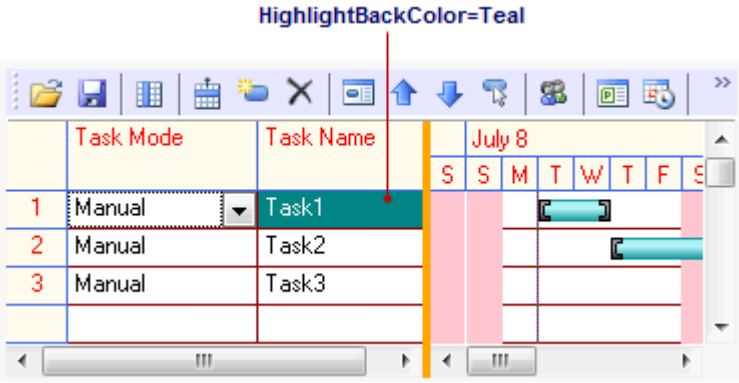
The following C1GanttView illustrates the **FixedForeColor** modified to **Red**.

Copyright © 2017 GrapeCity, inc. All rights reserved.



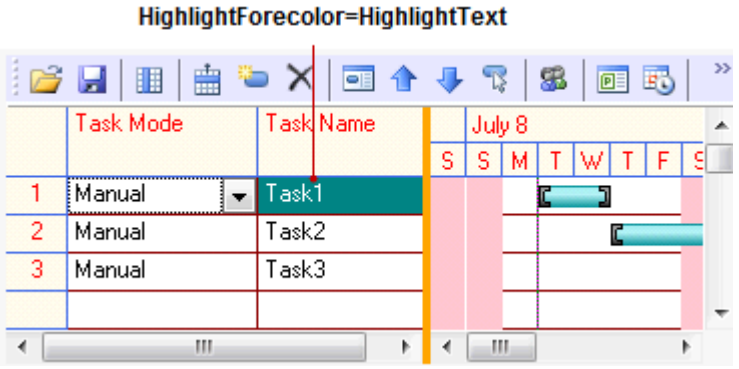
HighlightBackColor Example

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



HighlightForeColor Example

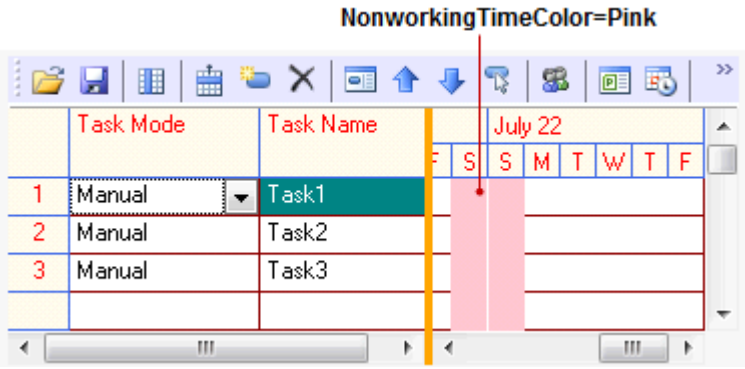
The following C1GanttView illustrates the **HighlightForeColor** modified to **HighlightText**.



NonworkingTimeColor Example

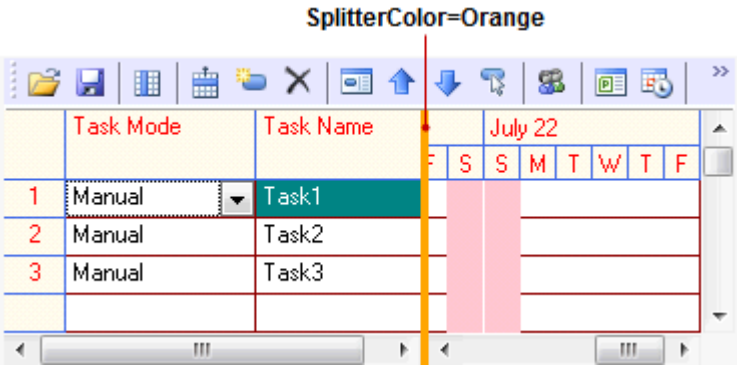
The following C1GanttView illustrates the **NonworkingTimeColor** property modified to **Pink**.

Copyright © 2017 GrapeCity, inc. All rights reserved.



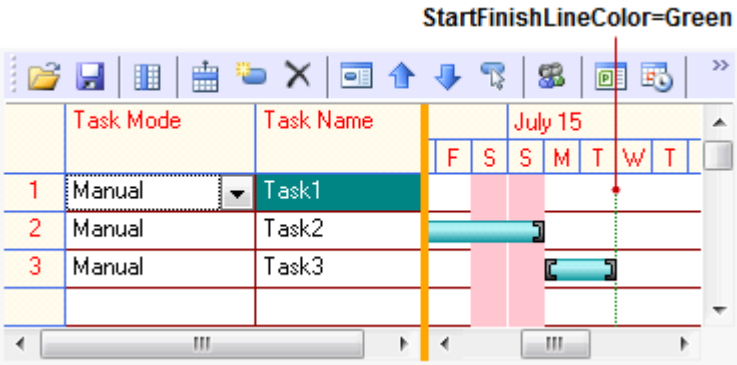
SplitterColor Example

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



StartFinishLineColor Example

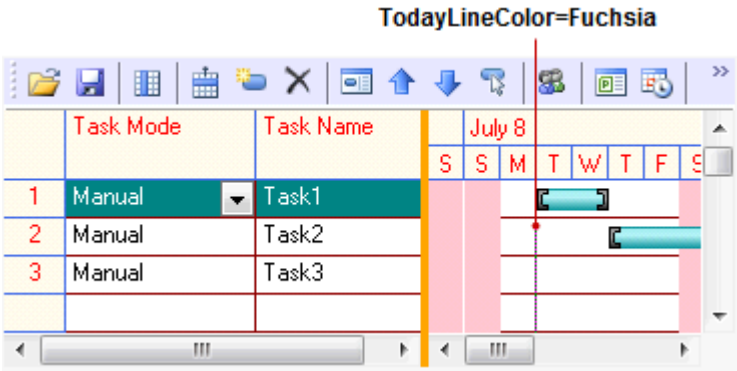
The following C1GanttView illustrates the **StartFinishLineColor** property modified to **Green**.



TodayLineColor Example

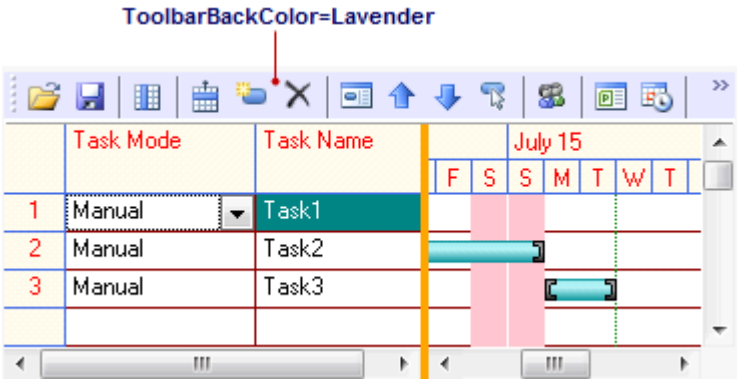
The following C1GanttView illustrates the **TodayLineColor** property modified to **Fuchsia**.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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

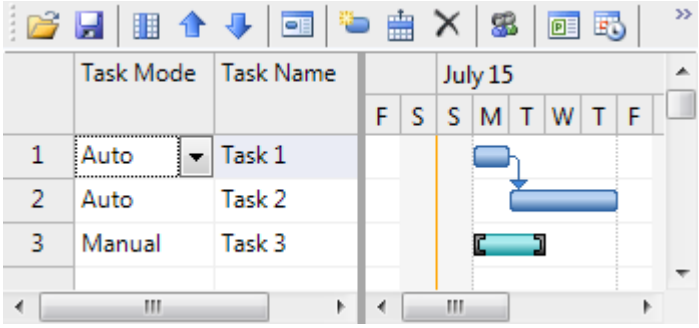
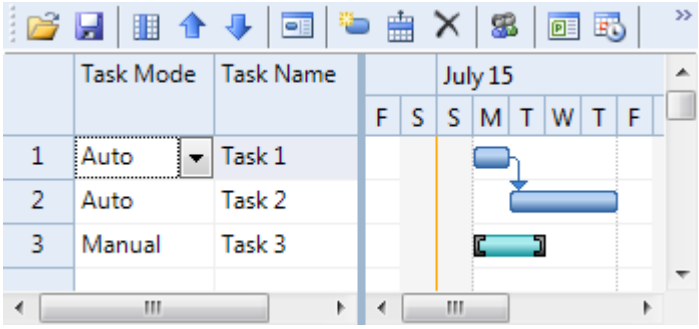
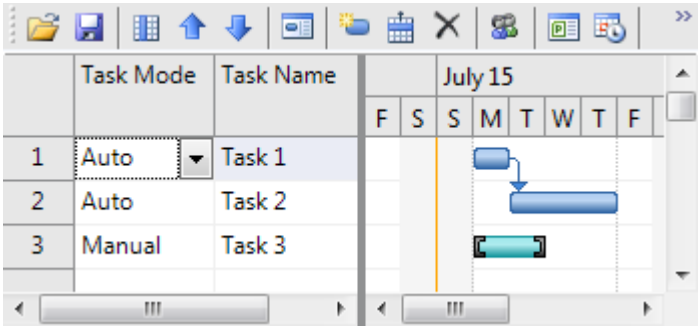
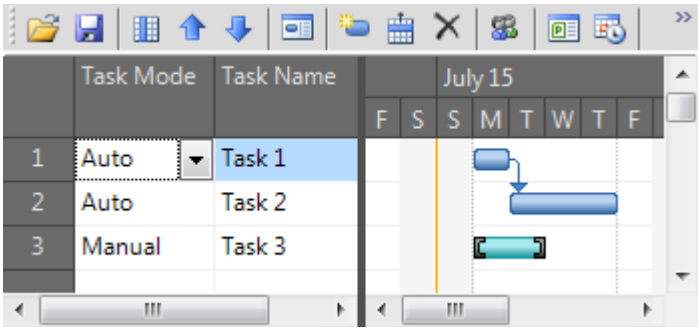
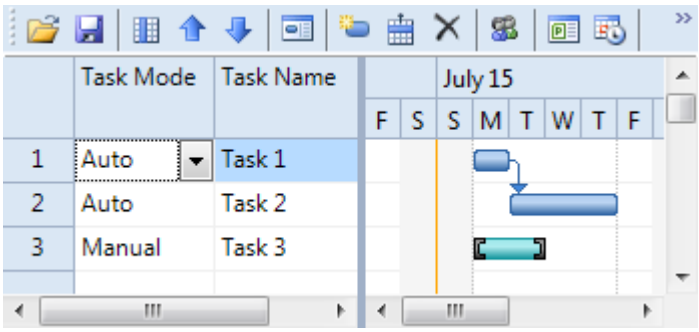
Visual Styles

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.

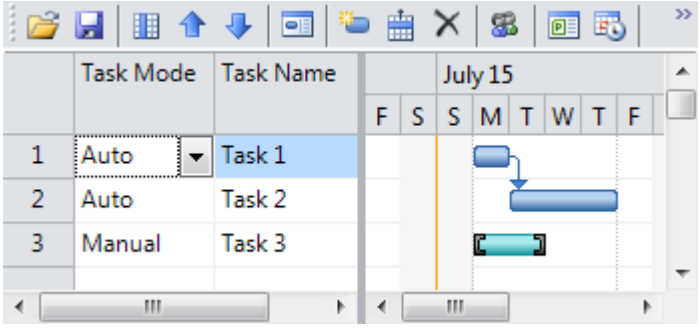
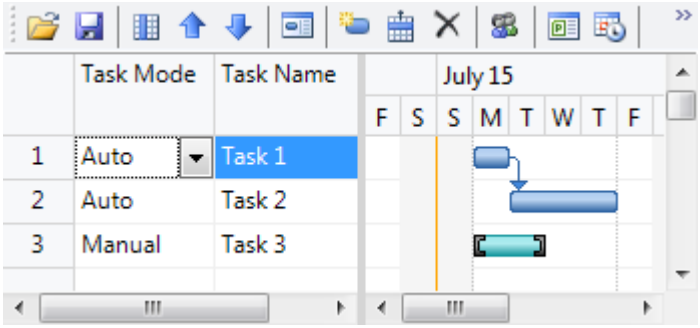
The following table illustrates each of the seven built-in visual styles:

Visual Style	Appearance

Copyright © 2017 GrapeCity, inc. All rights reserved.

Office2007Black	
Office2007Blue	
Office2007Silver	
Office2010Black	
Office2010Blue	

Copyright © 2017 GrapeCity, inc. All rights reserved.

Office2010Silver	
Windows7	

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.

For more information see Customizing the Bar Style.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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
- Task Mode
- Task Deadline
- Task Resources
- Task Start and Finish Time
- Task Constraints
- Task Predecessor
- Task Notes
- Milestones

Task Mode

C1GanttView uses two types of methods to schedule tasks: manual scheduling and 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.

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.

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.

Task Summary and Group

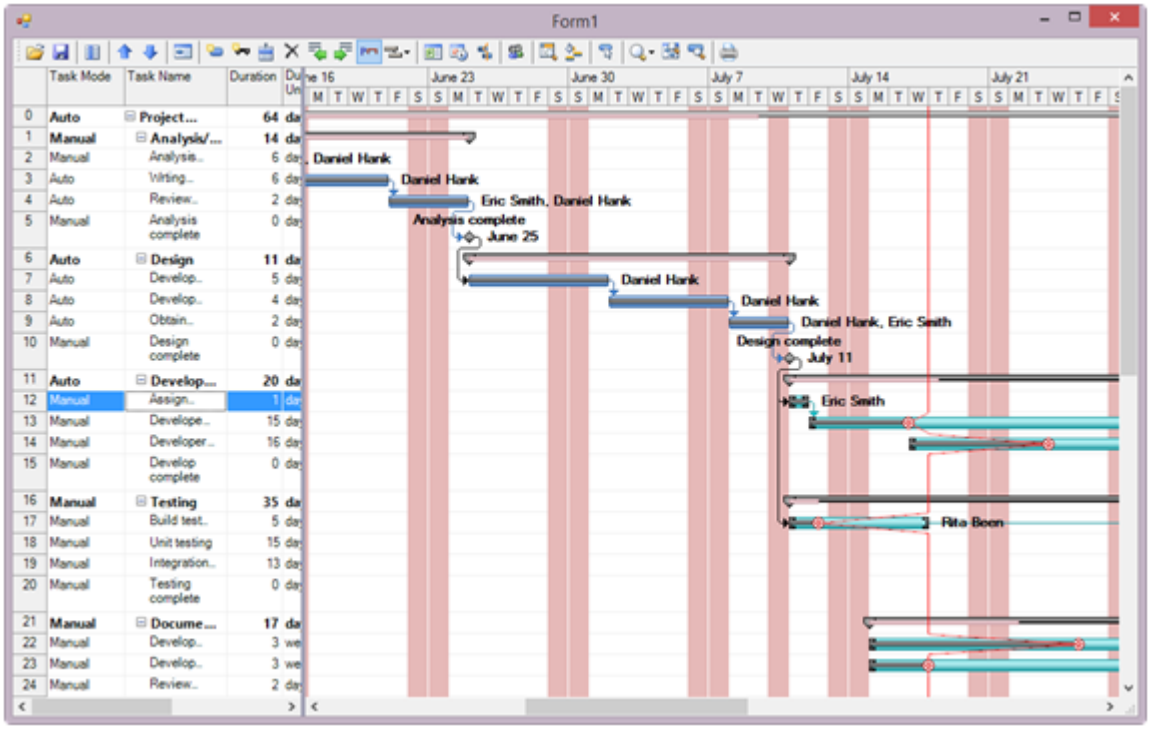
Task Summary

Copyright © 2017 GrapeCity, inc. All rights reserved.

The Summary task's appearance is determined by its children tasks which are the following properties:

- Task.Duration
- C1GanttView.StartDate
- C1GanttView.FinishDate
- C1GanttView.PercentComplete

The outline buttons in Task name columns of grid view provide ability to expand/collapse summary task. The outline structure is represented clearly in the Task Name columns of grid view. Tasks with different outline levels will have corresponding padding space. If any children tasks change properties, their parent tasks will be updated and appear similar to the following:



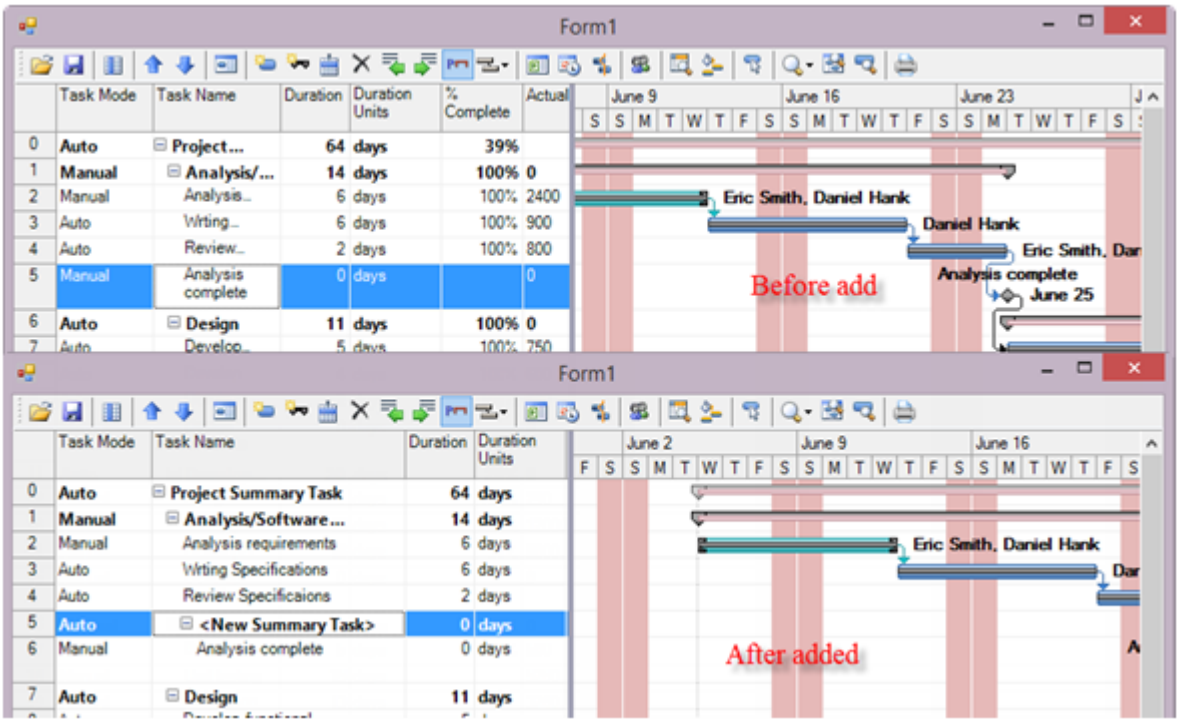
To distinguish the summary tasks from regular tasks you can customize the summary task font.

The tasks in gantt chart are represented as a tree structure, a task could be a regular or summary task. A summary task is an abstract task that some of their property values depend on all their children's properties. When a child changes its property values, its parent and ancestor task's properties will also be updated like the following:

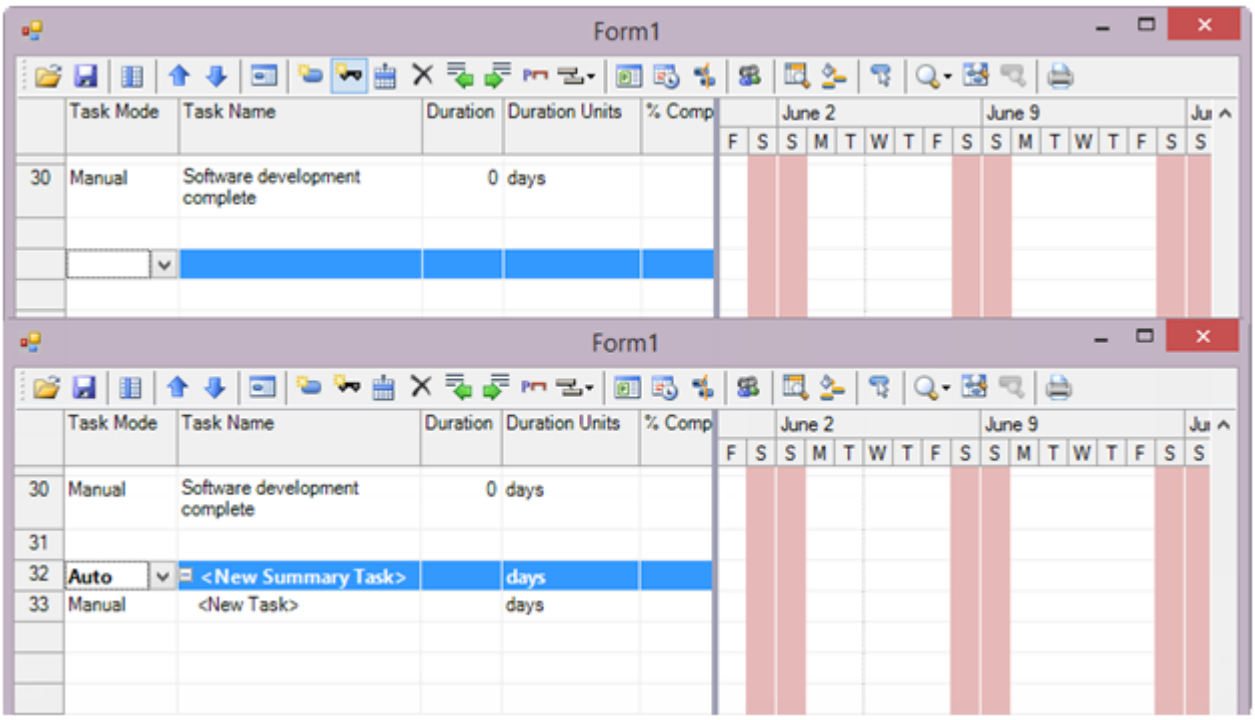
- StartDate - start date of a summary task is earliest start date of all its children task.
- FinishDate - finish date of a summary task is latest finish date of all its children task.
- PercentComplete - The percent complete of summary task is the total of finished time divide by total durations of its children.

When the selected task is initialized, the new summary task will be inserted before that task and adopt it as a new child. This new summary task also becomes the child of the old parent of that selected task and will appear like the following:

Copyright © 2017 GrapeCity, inc. All rights reserved.

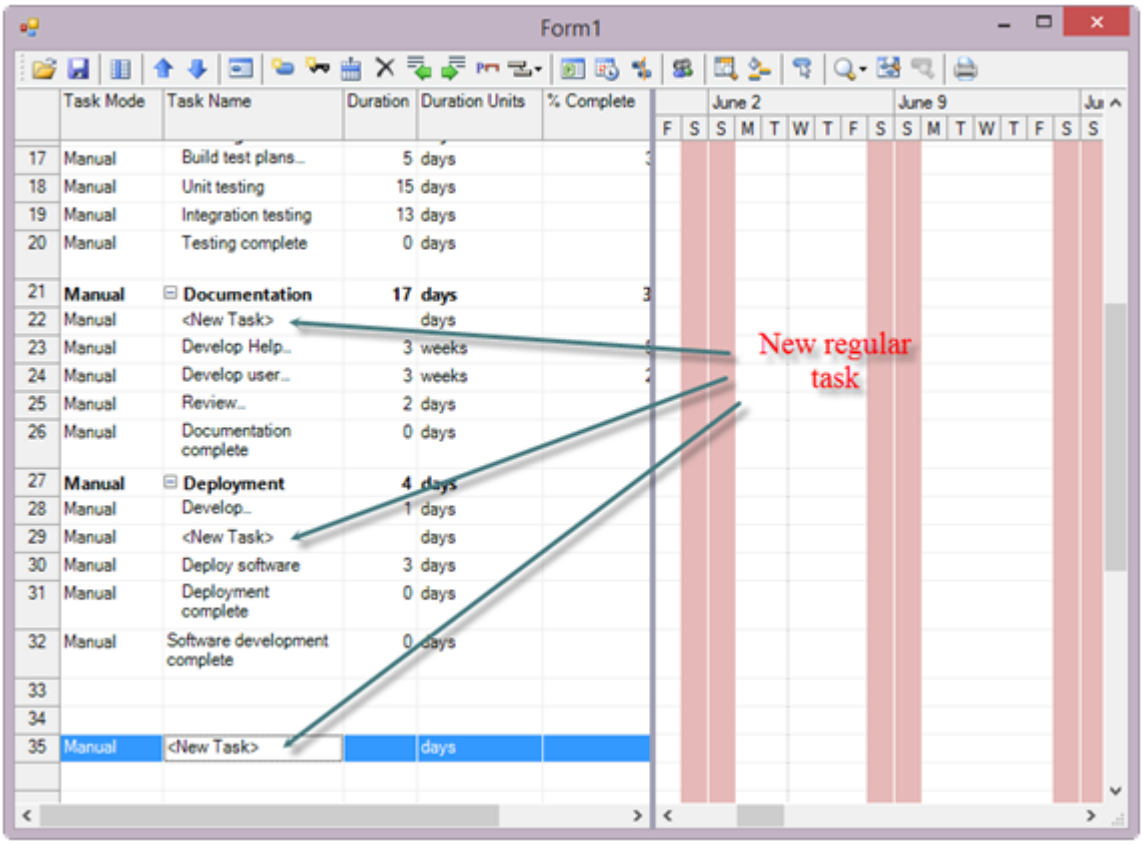


If the selected task is not initialized, a new child will be created then a new summary task will be added to that index as shown below:



The new task will be added at the selected task index and control will look up the previous tasks to find the nearest initialized task. If the found task is regular task, new task will be adopted as a child by found task's parent. Otherwise, new task will be adopted by this found task.

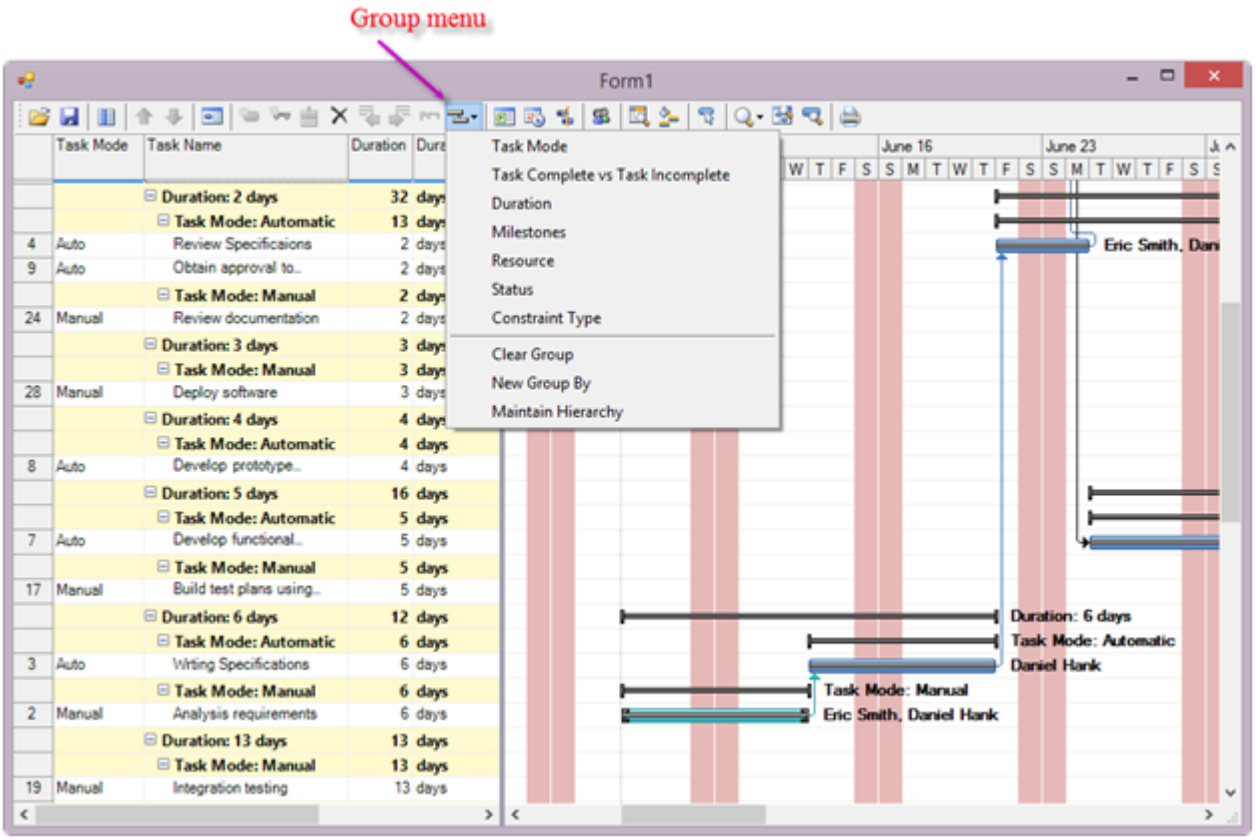
Copyright © 2017 GrapeCity, inc. All rights reserved.



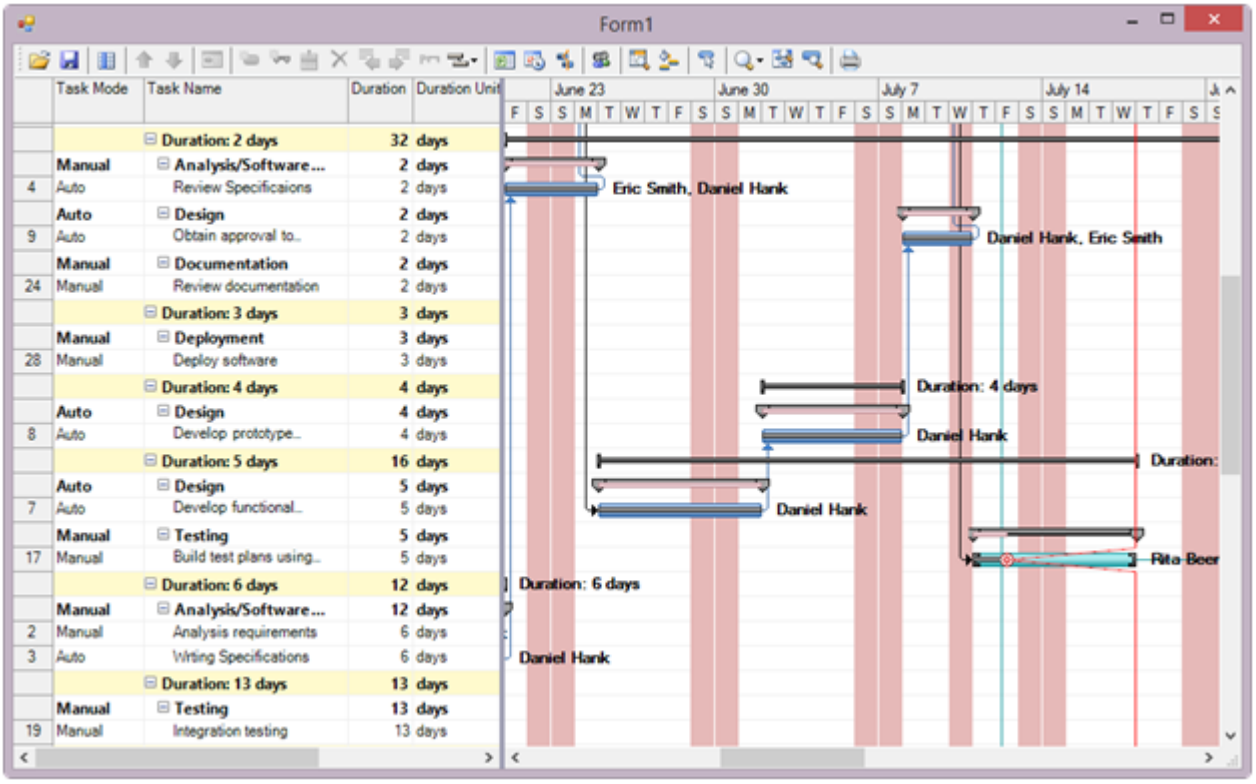
Task Group

By using the group feature, you can build an outline tree. Each group task is a summary task and all its children may have the same property values.
You can apply the group to an existing group to continue to classify its children.

Copyright © 2017 GrapeCity, inc. All rights reserved.



When grouping, the hierarchy from the normal view can be maintained like the following:



Group Behavior

Copyright © 2017 GrapeCity, inc. All rights reserved.

The regular task will be shown in group view. To distinguish group tasks from the regular tasks in grid view, you can change the **GroupBackColor** and **GroupFont** properties. You can also change the style for all group tasks by picking any group task and then changing the bar style for that task, other group task bar style will updated automatically.

When group view is presenting, the following tool strip items are disabled:

- Move task up/downAdd task
- Add Summary Task
- Add Blank Row
- Indent/Outdent Task
- Show Task Summary

If the selected task is group task or summary task, the following tool strip items are also disabled:

- Delete task
- Task Information (this item also doesn't appear when you right click to the task bar)

The following Group models are provided:

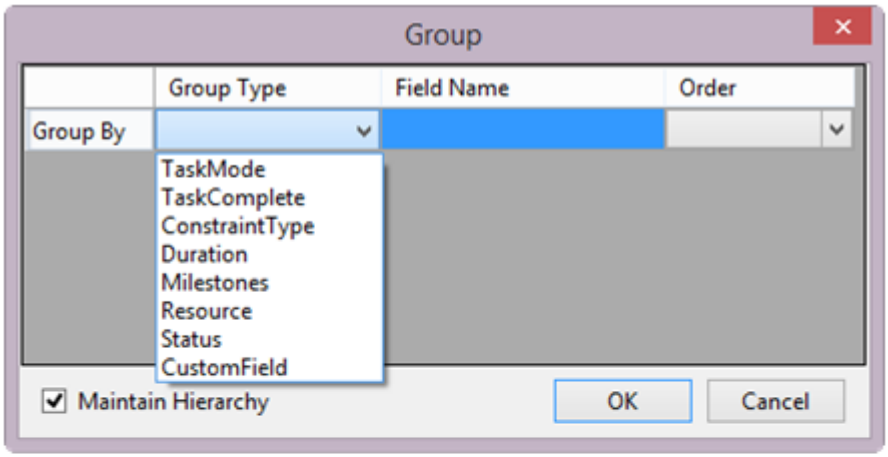
- Task Mode: Group the tasks by TaskMode property.
- Duration: Group the tasks by Duration property.
- Milestones: Group the tasks by two groups: milestone or not milestone.
- Resource: Group the tasks by their ResourceNames string.
- Status: Group the task by their PercentComplete and ProgressLine.
- ConstraintType: Group the task by their ConstraintType.
- Clear Group: Clear the group view, reset the gantt chart to regular view.
- New Group By: Invoke the Group form to advance group.
- Maintain Hierarchy: To show/hide the hierarchical structure in group view.

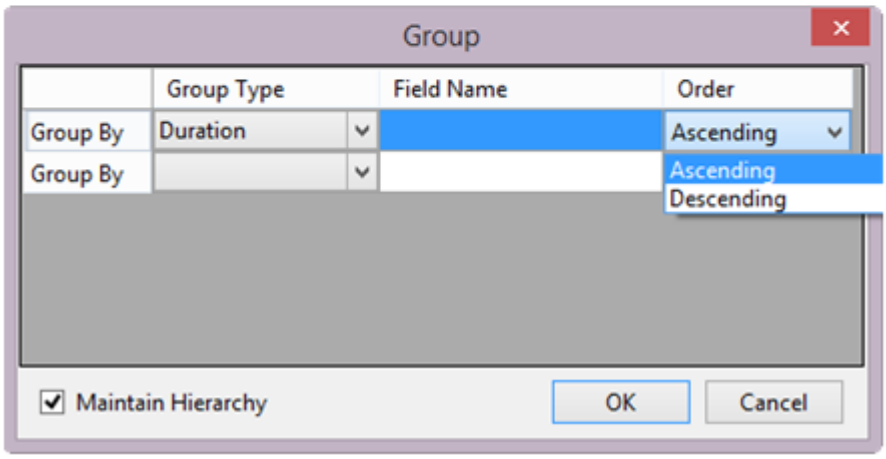
To create the advance group view, for example when we want to create a group view that contains two layers of group, open the **Group** form by selecting **New Group By** menu item from **Group** Menu. For each group layer, there're three fields to fill up in order to specify a group model:

- Group Type: The type of group such as: Duration, TaskMode, Status.
- Field Name: Only need to fill when Group Type is CustomField.
- Order: The order of group in gantt chart, default is Ascending order.

You could add any number of group models as you want, the blank group model will automatically add below the last group model in Group form each time you're done with creating the new group model.

Copyright © 2017 GrapeCity, inc. All rights reserved.





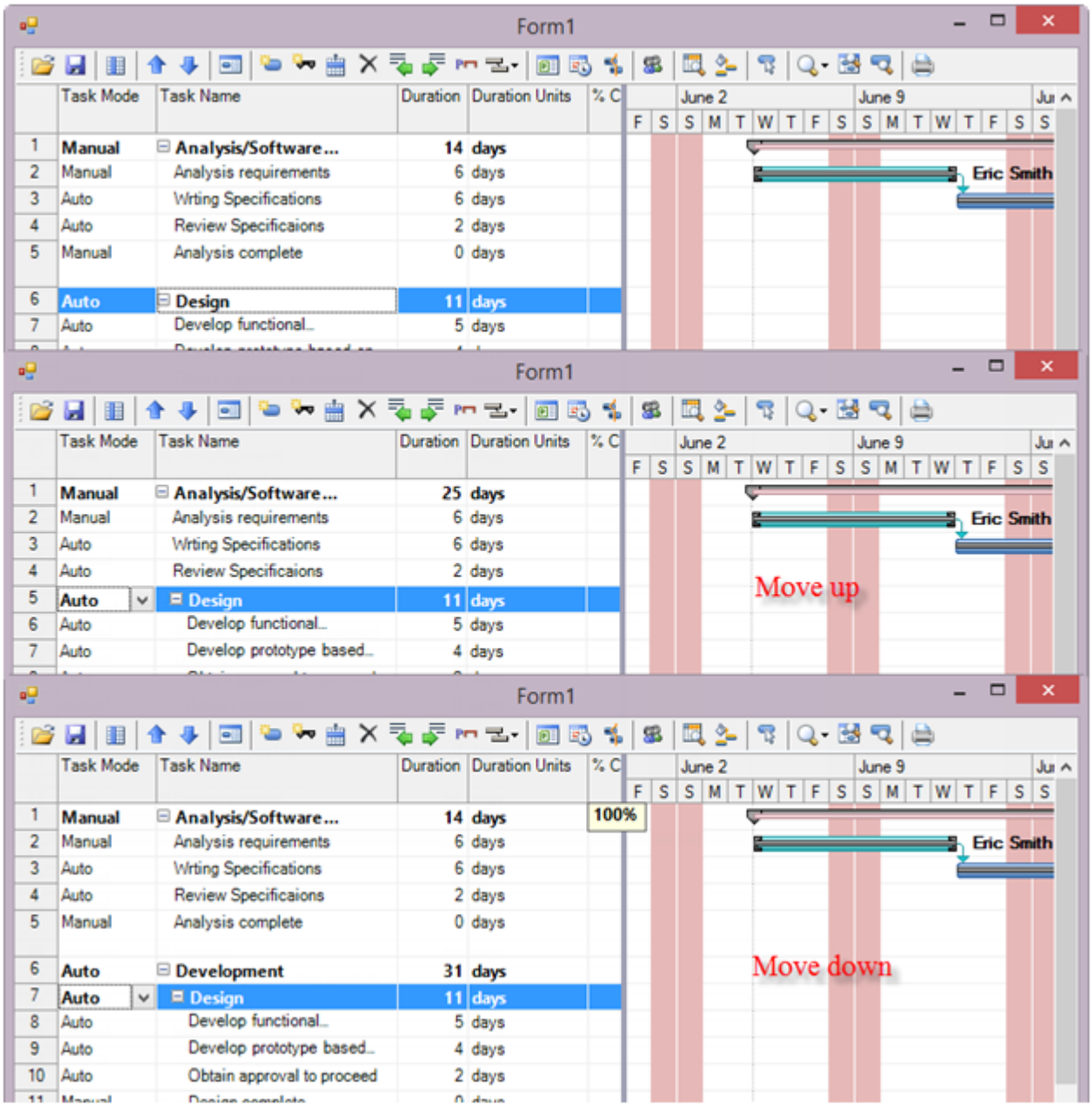
Deleting Summary Task

When deleting a summary task, it and all its children will be removed from gantt chart. If the deleted task is the last child of a summary task, that summary task will become a regular task.

Moving Task Summary Up and Down

After a task was moved up or down, if moved task is summary task, all its children will be moved also. If the previous/next task before move is summary task, moved task will be adopted as a new child by that task. Otherwise, the old neighbor's parent will adopted moved task.

Copyright © 2017 GrapeCity, inc. All rights reserved.



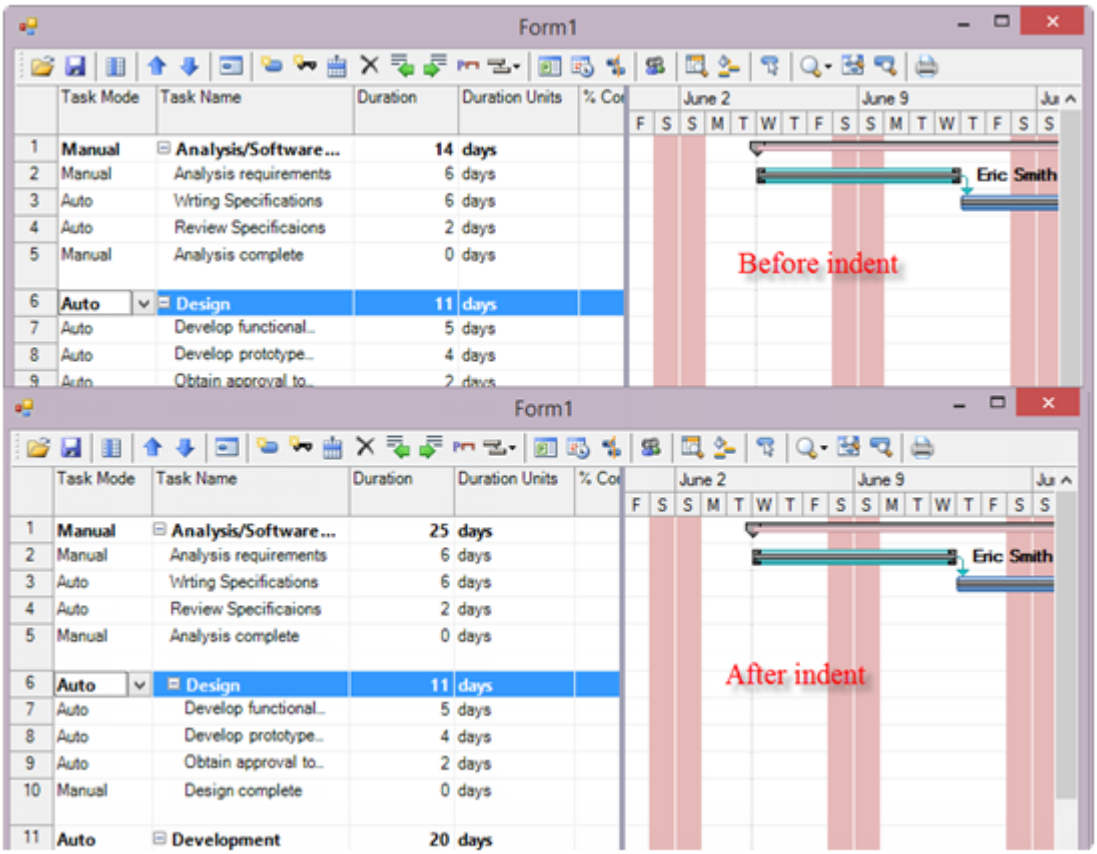
Indenting and Outdenting Summary Task

You could change the outline structure by using indent/outdenting buttons on the toolstrip. If the indented/outdented task is a summary task, all its children will be done, too. Things are simpler if the selected task is a regular task.

Indent Task

While the previous task has outline level greater than or equal to the selected task, the selected task are still able to indent. The indented task will be adopted by the upper nearest task that has the same outline level with it. If supposed parent is regular task, after indent the selected task, that parent will become a summary task

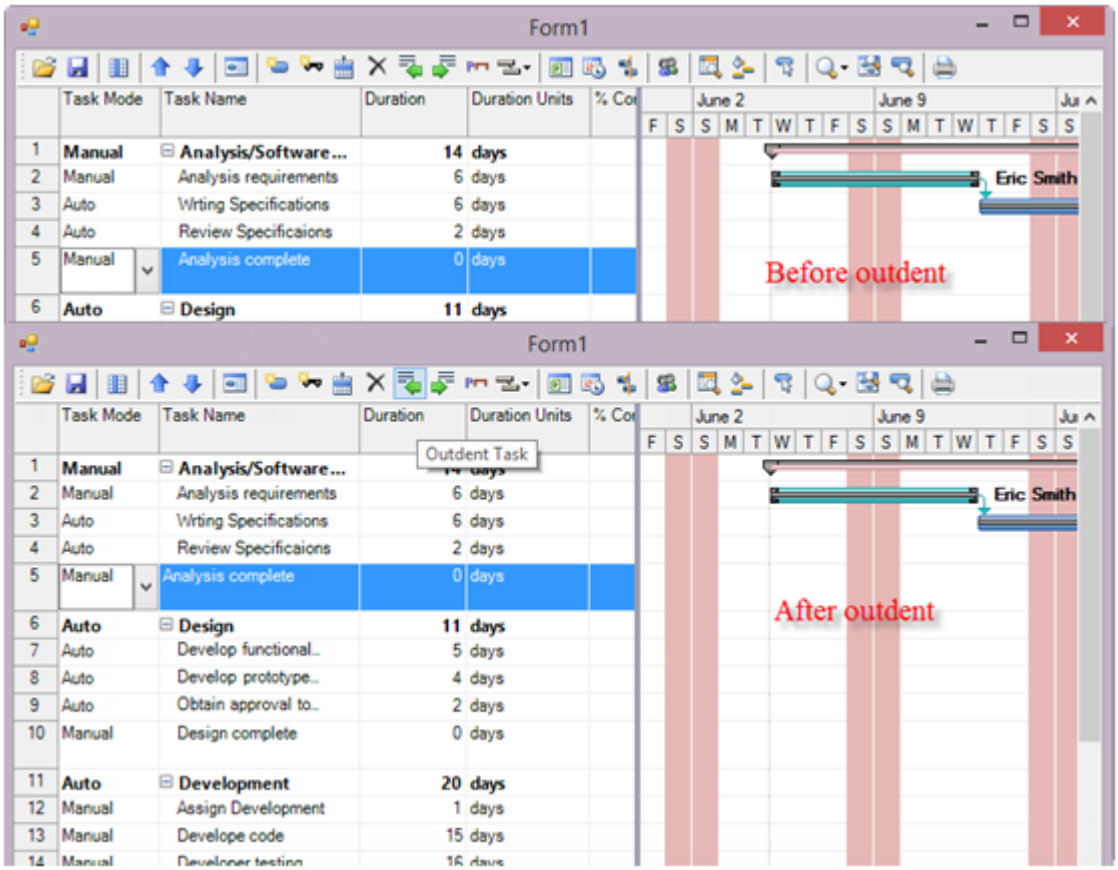
Copyright © 2017 GrapeCity, inc. All rights reserved.



Outdent Task

While the previous task has outline level less than or equal to the selected task, the selected task are still able to outdent. The outdented task will be adopt as a new child of its grandparent. If it is the last child of its parent before outdent, after that, the old parent will become a regular task.

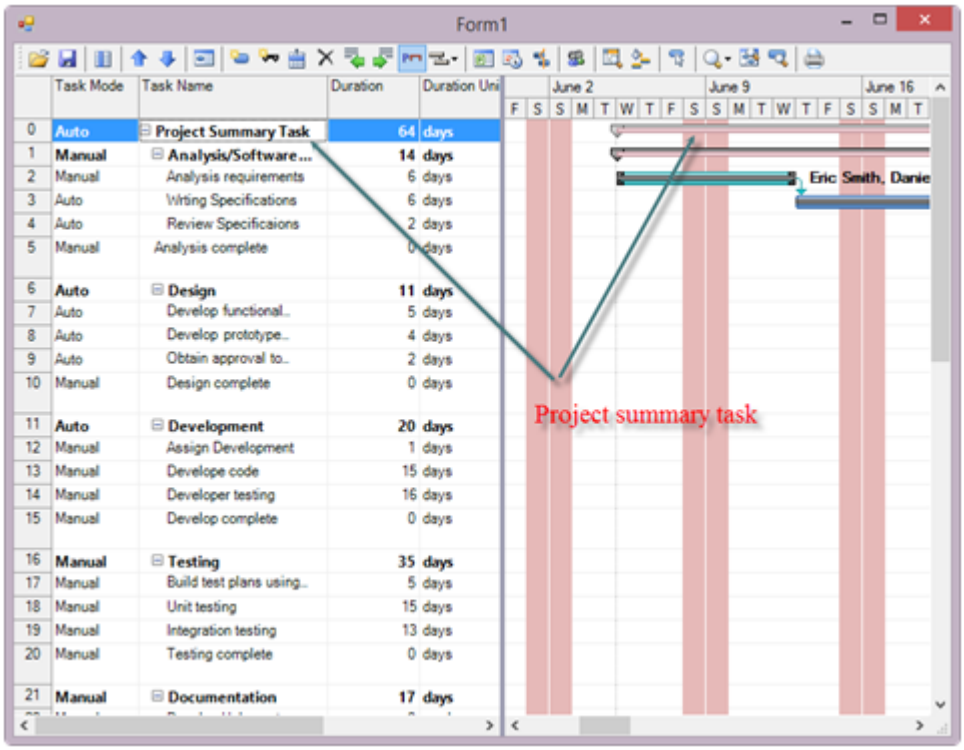
Copyright © 2017 GrapeCity, inc. All rights reserved.



Showing and Hiding Summary Tasks

You can show or hide project summary tasks by clicking the Show summary button from the toolstrip. By default, this task is hidden from the gantt chart.

Copyright © 2017 GrapeCity, inc. All rights reserved.

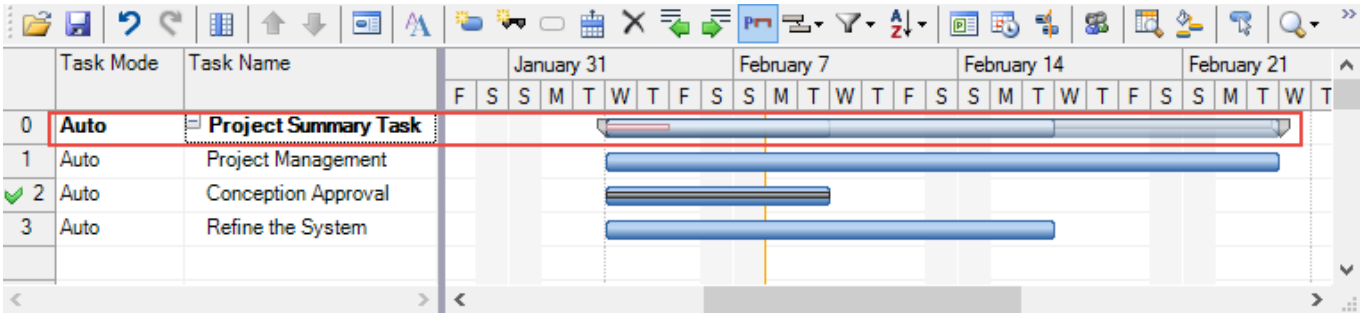


Showing Markers in Project Summary Task Bar

You can represent multiple tasks on the Project Summary Task bar by using ReflectOnSummary property for each task. Set the **ReflectOnSummary** property to True for the tasks that you want to be shown on the Project Summary Task bar. The tasks are then represented on the bar through markers or outlines that differentiate multiple tasks.

You can also select the **Reflect on summary bar** option for the tasks at design time on the **Task Information** dialog box.

The following image shows an example of multiple tasks represented on the Project Summary Task bar:



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.

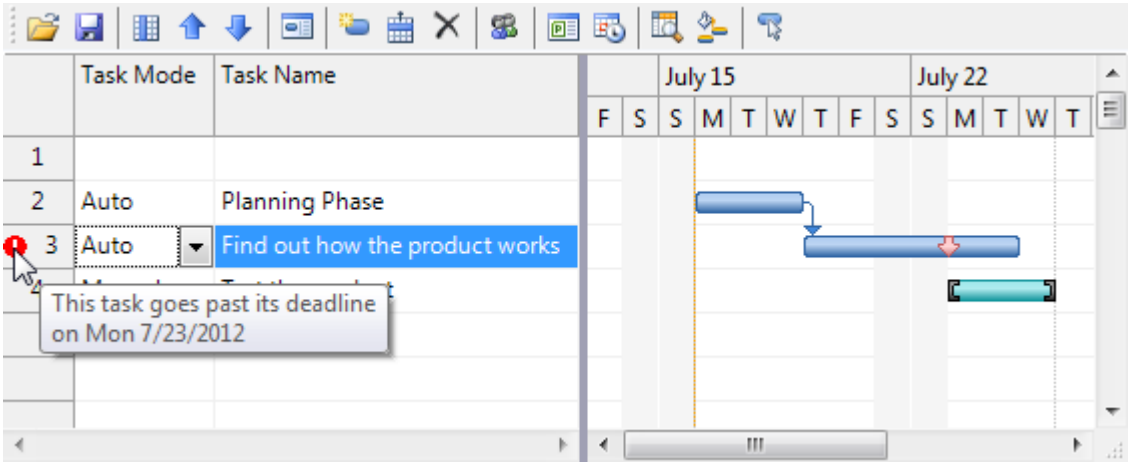
Copyright © 2017 GrapeCity, inc. All rights reserved.

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,

Copyright © 2017 GrapeCity, inc. All rights reserved.

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.

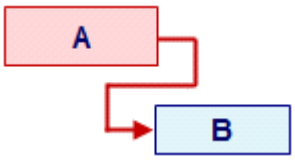
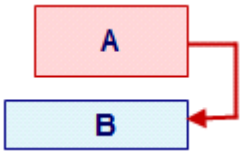
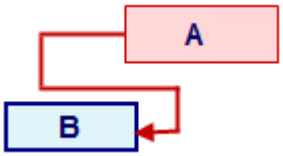
Task Predecessor

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.

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 C1GanntView. 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 (A) is finished.	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.
	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 book shelves for your construction project are built off-site. Two of the tasks in your project are

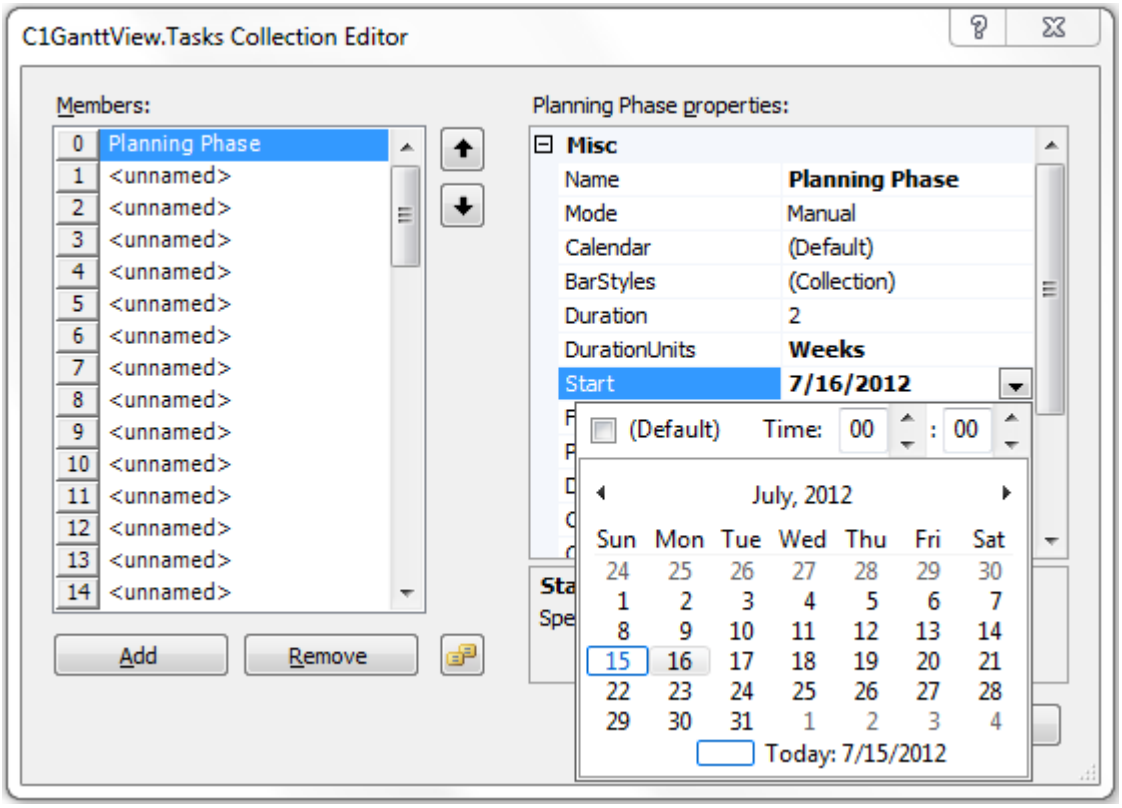
Copyright © 2017 GrapeCity, inc. All rights reserved.

			"Wood delivery" and "Assemble book shelves". The "assemble bookshelves" task cannot be completed until the "wood delivery" task begins.

Note: Use the Predecessors property to identify the predecessors of the given task. However, to determine whether a task is predecessor to some other task use the Successors property. This gets all the successors of a given task.

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 send 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 send 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:

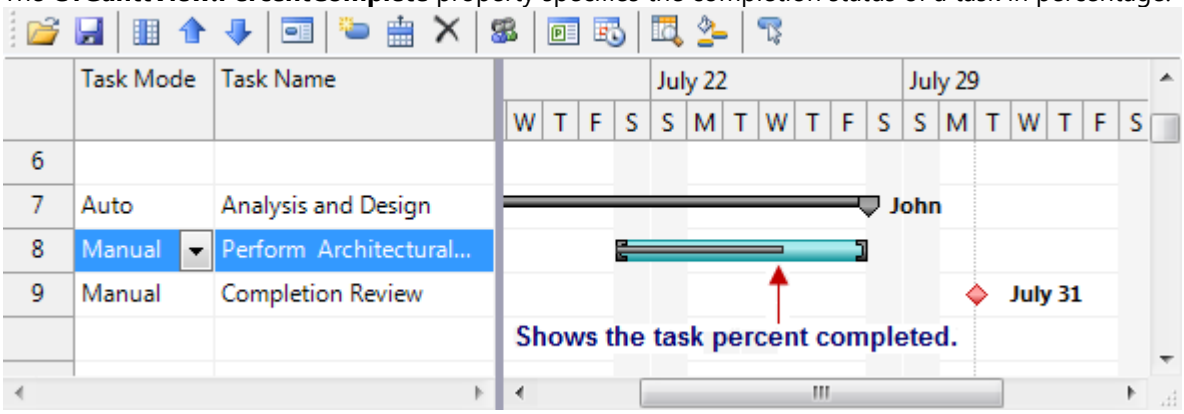
- Unsure when the project begins
- Would like to find out when the project should start by first entering a specific finish date

Copyright © 2017 GrapeCity, inc. All rights reserved.

- Required to schedule the project from a finish date

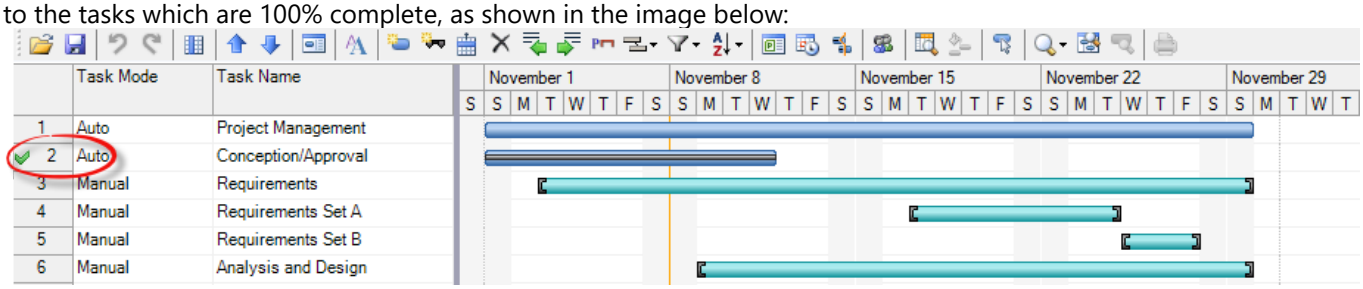
Task Percent Complete

The **C1GanttView.PercentComplete** property specifies the completion status of a task in percentage.



Note: In code, value of PercentComplete property should be between 0 and 1. For example, if you want to set PercentComplete property for task 3 to 30% then, write the following code:
c1GanttView1.Tasks[2].PercentComplete= 0.3;

A 100% complete task is represented differently in the C1GanttView control. A task completion indicator appears next to the tasks which are 100% complete, as shown in the image below:



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.

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

Copyright © 2017 GrapeCity, inc. All rights reserved.

to the left, right, top, or bottom of the milestone marker.

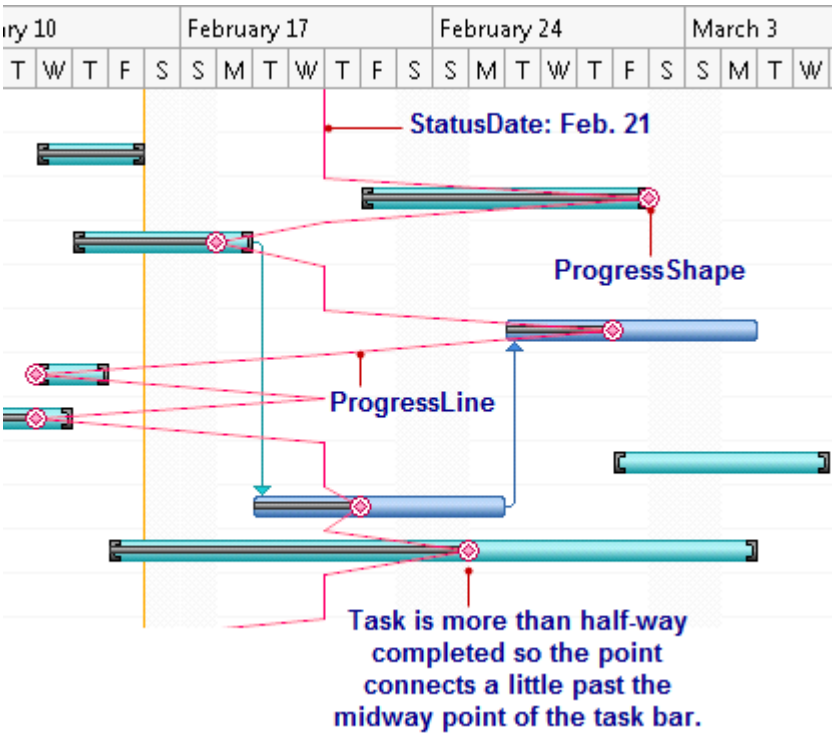
See Creating a Milestone for more information.

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

Print Overview

The Printing feature can be used when you do the following:

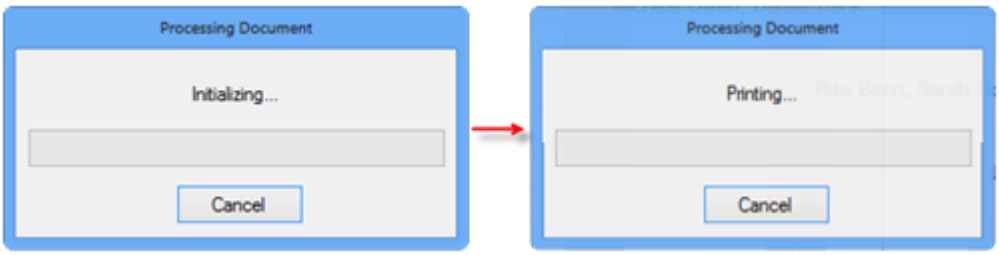
- Add a reference to the C1.C1Report and C1.Win.C1Report assemblies to make the Print Preview function available. The Printing feature uses the C1PrintDocument and C1PrintPreviewDialog in **C1.C1Report** and **C1.Win.C1Report** respectively, to print and invoke the preview dialog.
- Set the the C1GanttView.EnablePrinting property is set to **True**. Once this property is enabled, the **Print** button, , automatically appears enabled on the C1GanttView toolbar and can be used at runtime.

C1GanttView project includes a PrintInfo class to control the information related to printing the schedule. The properties in the PrintInfo class are available at design time, run time, and programatically. At run time these properties appear in the Print dialog box where you can select the following:

- Printing Style
- Specify Settings for Header, Footer, and Legend
- Page Setup
- Print Range
- Print
- Print Preview

When printing the following steps are typically used in this order:

1. Click the **Print** button which will open the Print dialog box.
2. Select the type of print layout from the prints styles in the Print dialog box.
3. Click the **Settings** button in the **Print** dialog box to open the Style Settings dialog box where you can customize the header, footer, and legend appearance for each page.
4. Click the **Page Setup** button so you can specify the page's size, source, orientation, and margins.
5. Determine the print range. You can select a specific range to print by clicking the radio button **Print specific dates** and select start and end date in the two combo boxes.
6. Click the **OK** button in the Print dialog box to start printing by default printer or click **Preview** button to review the printing appearance. Once the print progress is starting, the **Processing Document** form will appear.



The **Processing Document** shows the status and the percentage complete of progress. It also provides a **Cancel** button to cancel the operation. The print operation will use system default printer to print.

When the preview progress is starting, the **Processing Document** form will appear to shows the status of initializing progress. During this time the preview document is generated, then the Print Preview dialog will appear. In the Print Preview dialog box you can change page setup, zoom, save, print...etc.

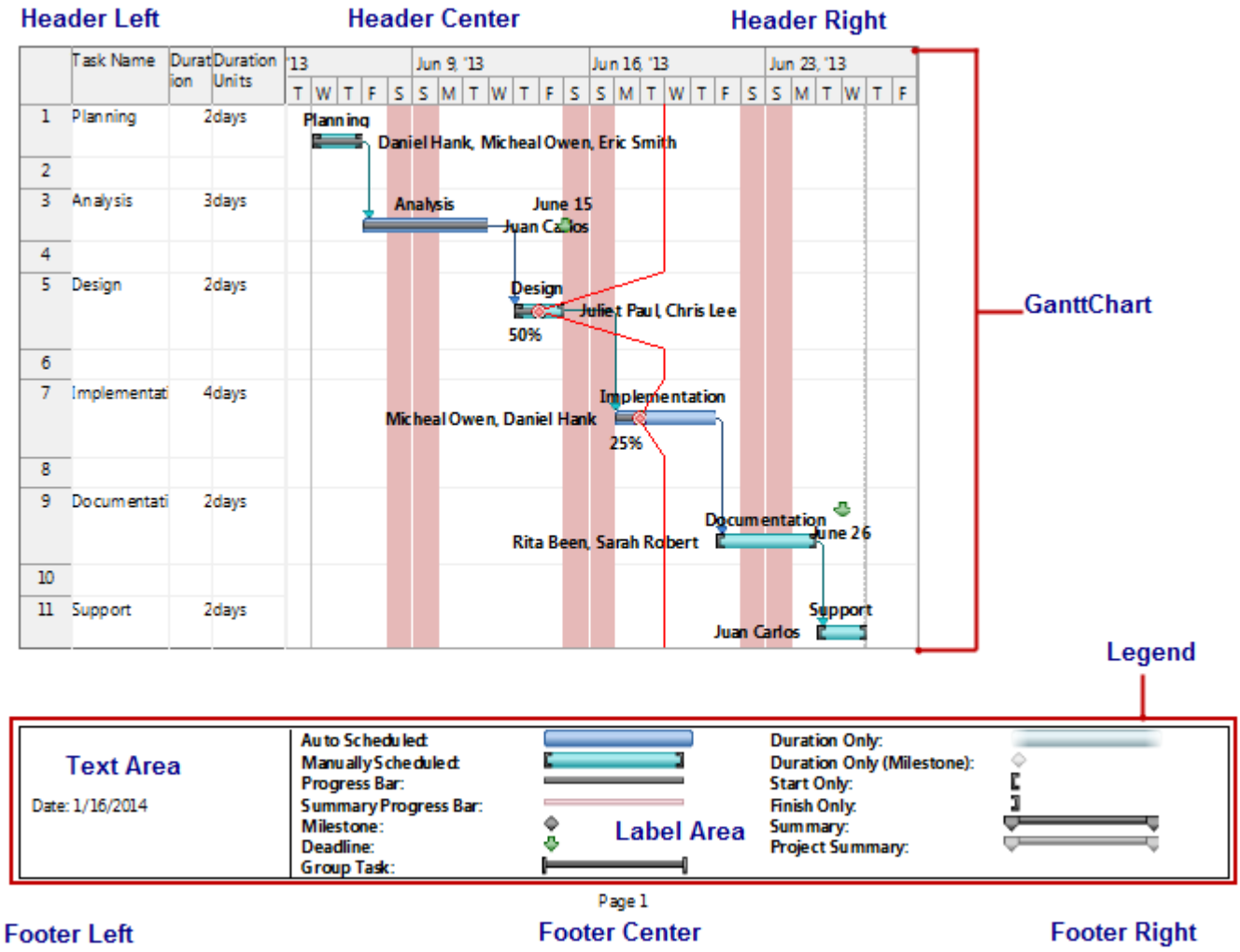
Copyright © 2017 GrapeCity, inc. All rights reserved.

Print Layout

The print layout for the C1GanttView project is determined by the following elements:

- Header/Footer - Divides the GanttView into three parts. Each part represents a column. You can add content and format it for each part.
- GanttChart- Includes the GridView and ChartView; it represents all task information and graphics representation.
- Legend - Represents various patterns of bar styles that have been used in the project as well as an annotation area to look up in the ChartView.

The following GanttChart illustrates the printing elements used for the print layout:



Copyright © 2017 GrapeCity, inc. All rights reserved.

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 Studio.

Please refer to the pre-installed product samples through the following path:

Documents\ComponentOne Samples\WinForms

The following table provides a short overview of each sample.

Sample	Description
CustomColumn	This sample adds the special column (Actual Cost) that is calculated as a full cost of the resource multiplied by the amount of the resource usage. The value in the "Actual Cose" column is recalculated each time when any of its source values (such as resource cost or task duration) is changed.
HowToDoItInCode	This sample shows how to do runtime things in code such as loading and saving ganttview files, adding predecessors, adding resources, adding time scales, removing time scales, inserting tasks, removing tasks, and deleting tasks.
PrintingSample	This sample represents how the printing feature works with the custom styles.
SoftwareDevelopmentPlan	Demonstrates an example that uses C1Ribbon to replace ToolStip in C1GanttView.
ZoomSample	This sample shows how to do some zoom operations using the C1GanttView control.

Copyright © 2017 GrapeCity, inc. All rights reserved.

GanttView for WinForms Task-Based Help

The task-based help section assumes that you are familiar with programming in the Visual Studio environment and have a general understanding of the **C1GanttView** 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.

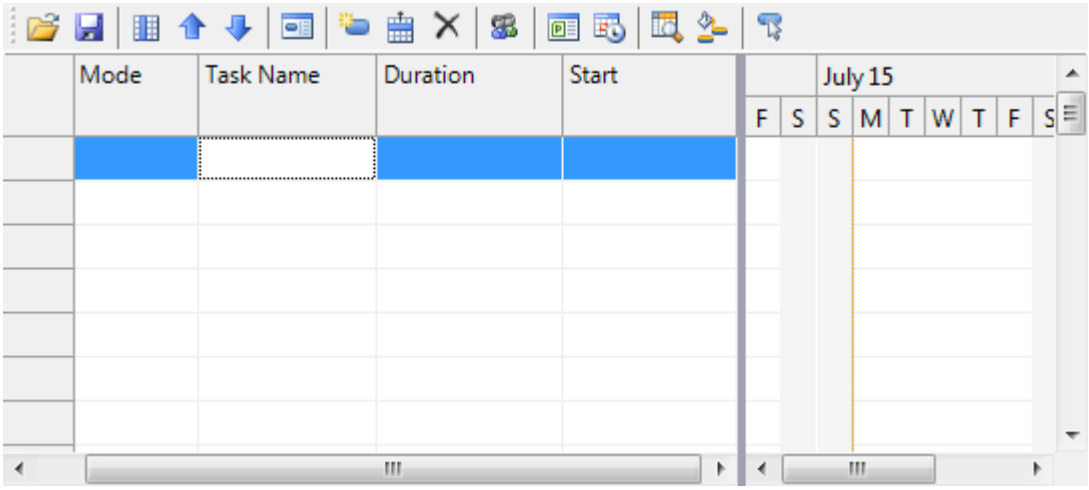
Adding Columns to the Grid

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

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**:

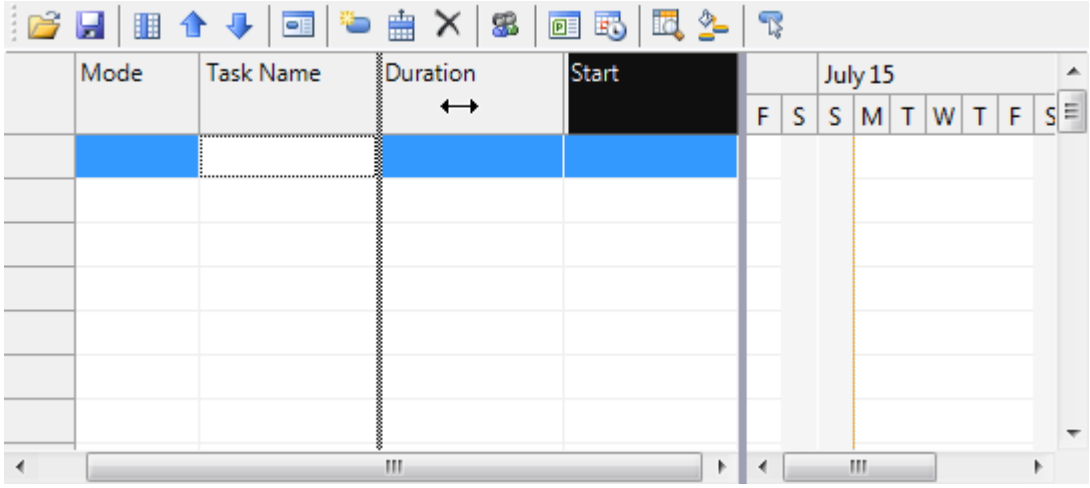


Moving Columns in the Grid

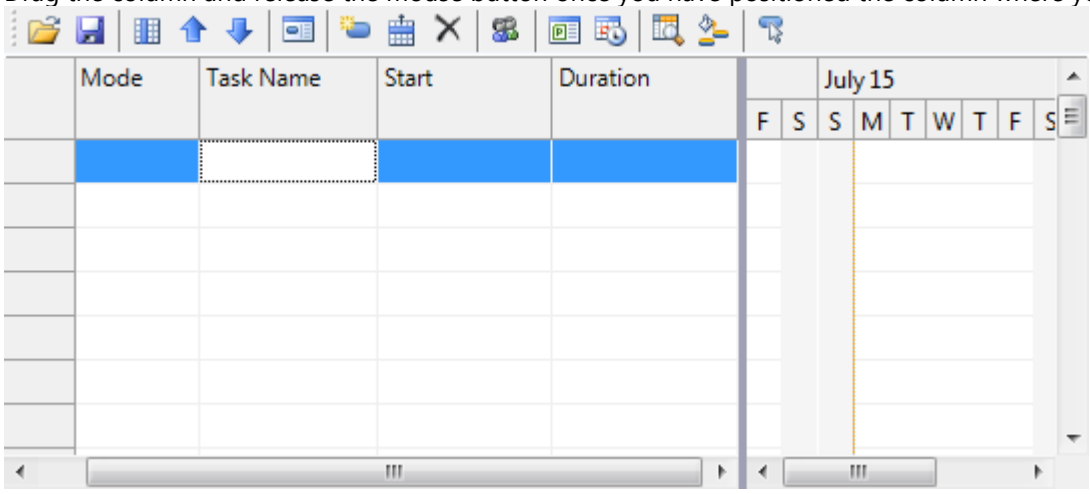
To move a column in the grid at run time complete the following:

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

Copyright © 2017 GrapeCity, inc. All rights reserved.



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



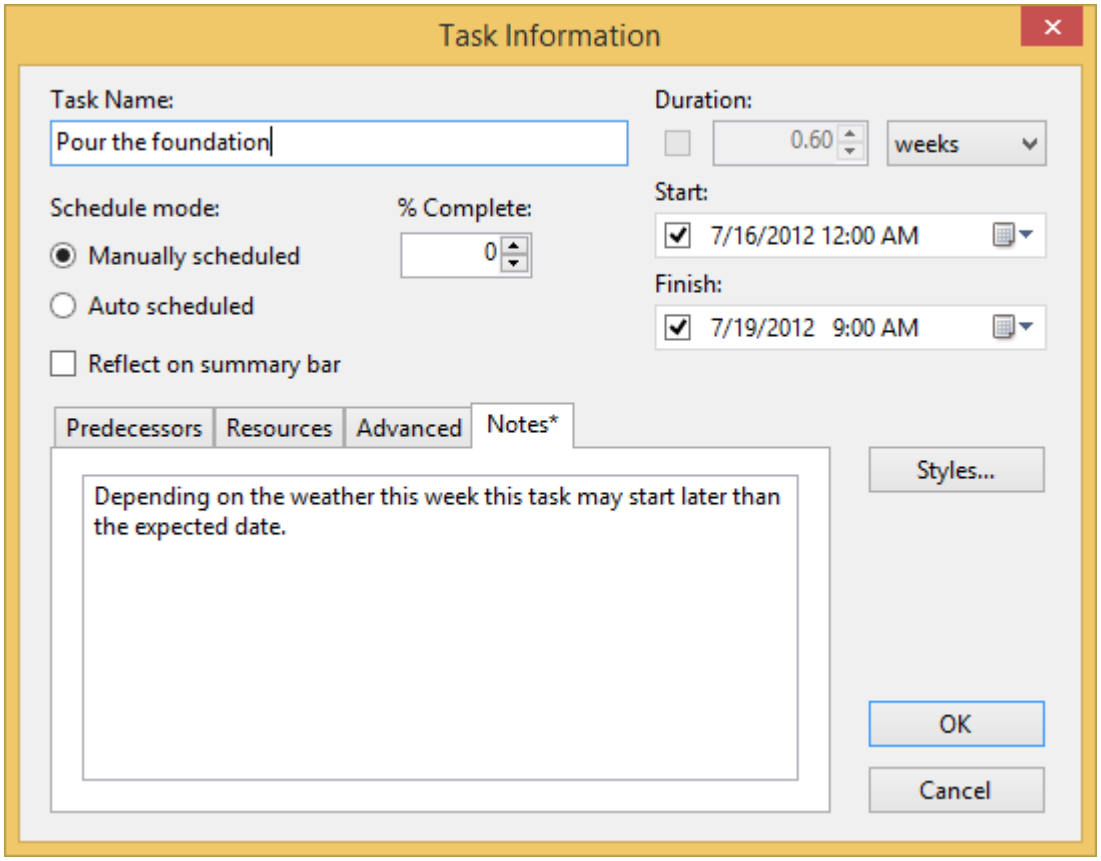
The **Start** column appears before the **Duration** column.

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.

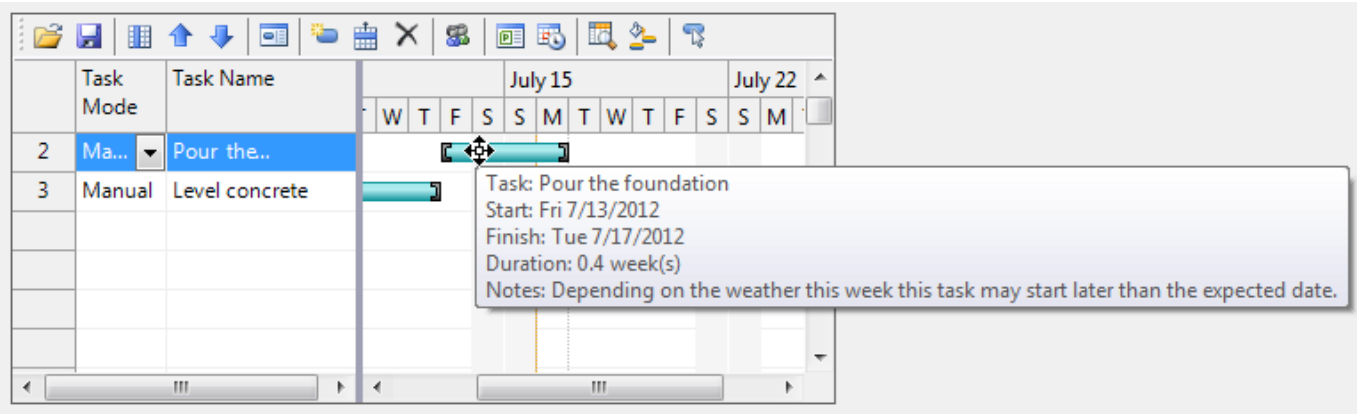
Copyright © 2017 GrapeCity, inc. All rights reserved.



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.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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:

To write code in Visual Basic

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
```

To write code in C#

C#

```
private void btnAddResource_Click(object sender, EventArgs e)
{
    // add the new Resource object
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
    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);
    }
}
```

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.

To write code in Visual Basic

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
```

To write code in C#

C#

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

Removing Top Level From the Time Scale

To programatically remove the top level from the time scale, complete the following:

1. Add a button control to your form above the C1GanttView control.
2. Name your button control to **btnRemoveTopLevel** and add the following event hander to your button control:

Copyright © 2017 GrapeCity, inc. All rights reserved.

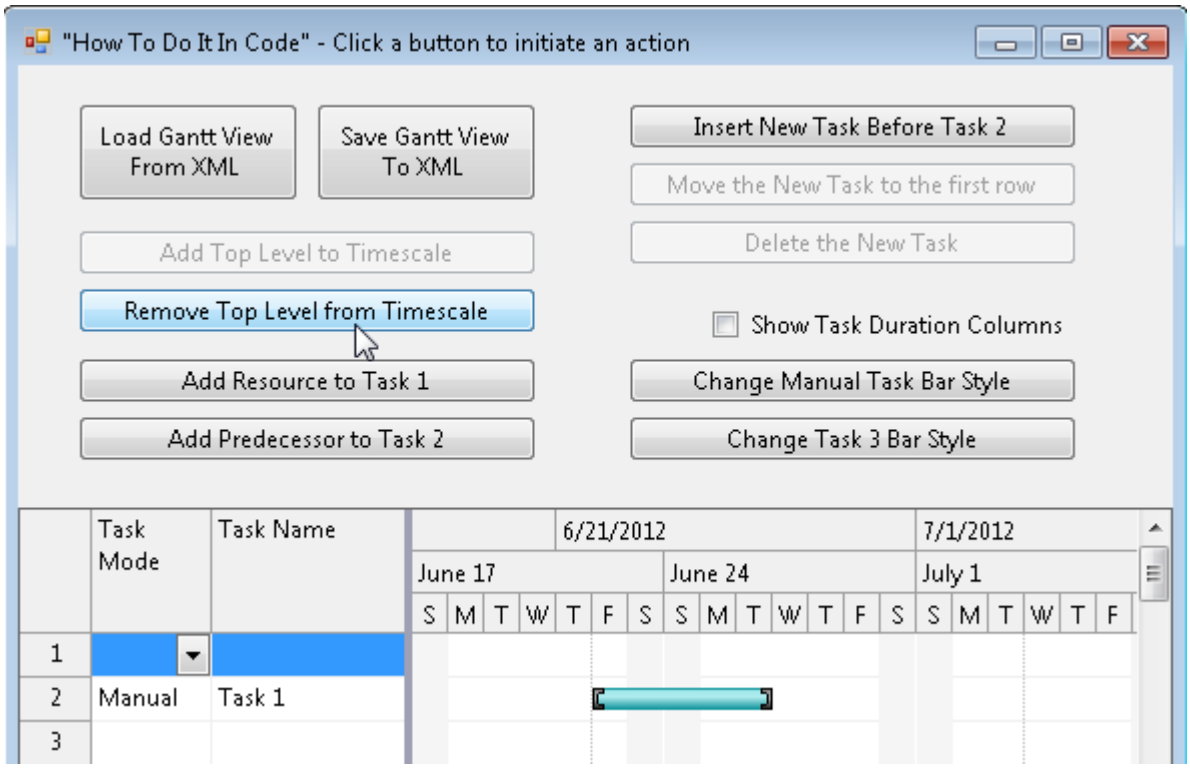
Visual Basic

```
Private Sub btnRemoveTopLevel_Click(sender As Object, e As EventArgs)
        gv.Timescale.TopTier.Visible = False
        btnAddTopLevel.Enabled = True
        btnRemoveTopLevel.Enabled = False
End Sub
```

C#

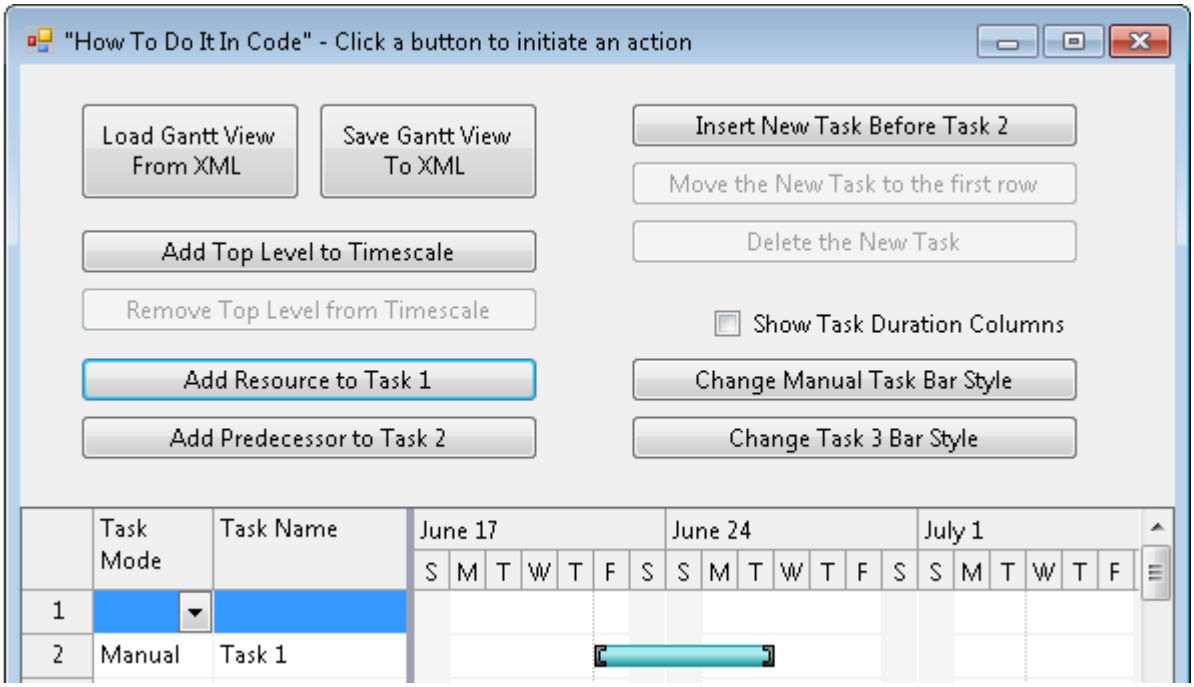
```
private void btnRemoveTopLevel_Click(object sender, EventArgs e)
  {
       gv.Timescale.TopTier.Visible = false;
       btnAddTopLevel.Enabled = true;
       btnRemoveTopLevel.Enabled = false;
  }
```

3. Run your project and click on the **Remove Top Level** from Timescale button.



4. The Timescale Tier is removed from the top of the tier.

Copyright © 2017 GrapeCity, inc. All rights reserved.



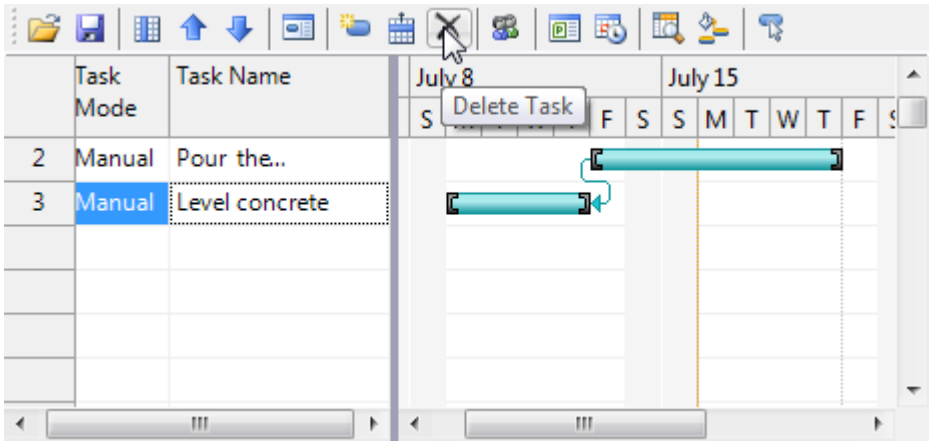
For more information on this task, see the sample **HowToDoItInCode**.

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:

Copyright © 2017 GrapeCity, inc. All rights reserved.

To write code in Visual Basic

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
```

To write code in C#

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();
    }
}
```

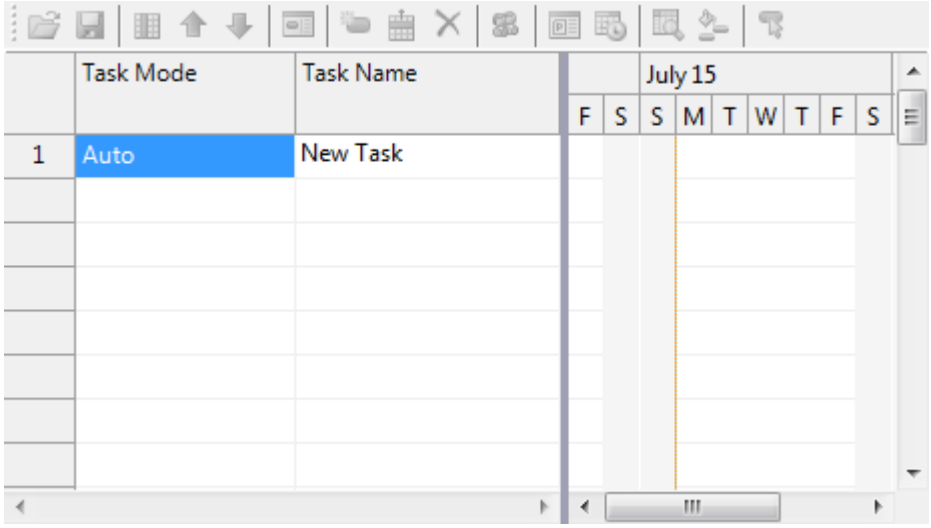
Inserting a Task

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.

Copyright © 2017 GrapeCity, inc. All rights reserved.



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:

To write code in Visual Basic

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

To write code in C#

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);
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

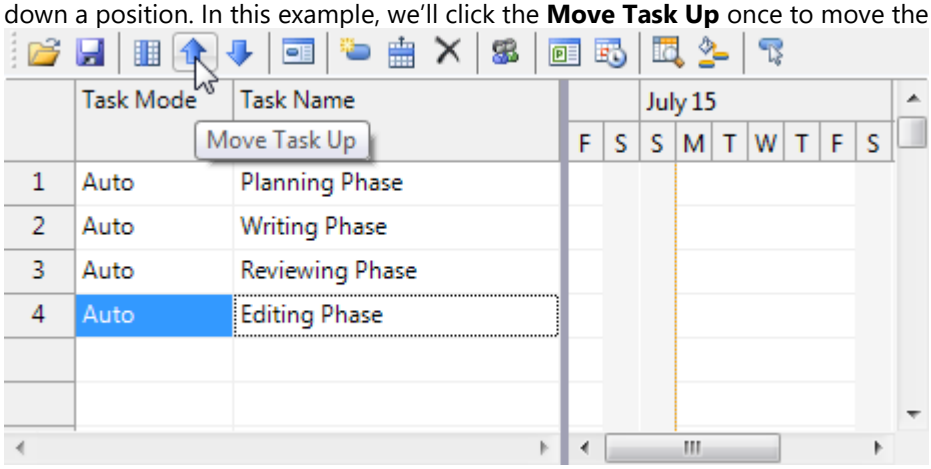
```
            t.Mode = TaskMode.Automatic;
            t.Name = "New Task";
            t.Start = new DateTime(2012, 6, 25);
            t.Duration = 3;
        }
    }
```

Moving a Task

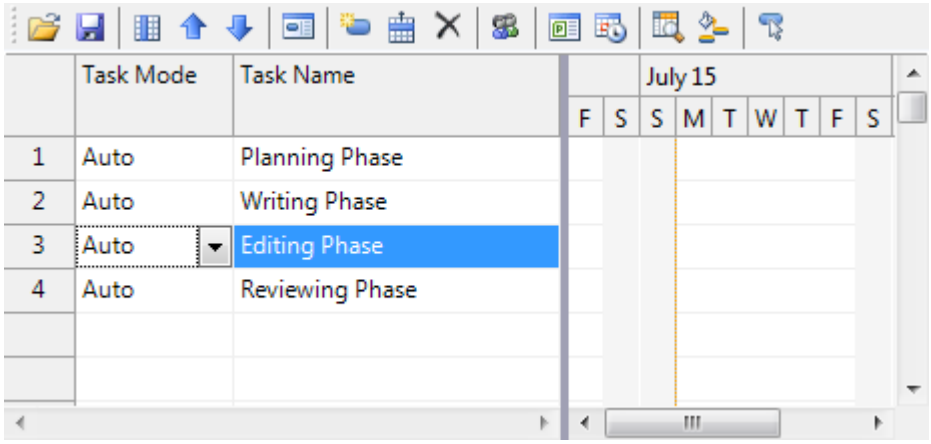
You can move a task at run time using the up or down arrows located on the C1GanttView toolbar or you can move tasks programmatically using the **RemoveAt** and **Insert** methods.

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:

To write code in Visual Basic

Copyright © 2017 GrapeCity, inc. All rights reserved.

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
```

To write code in C#

C#

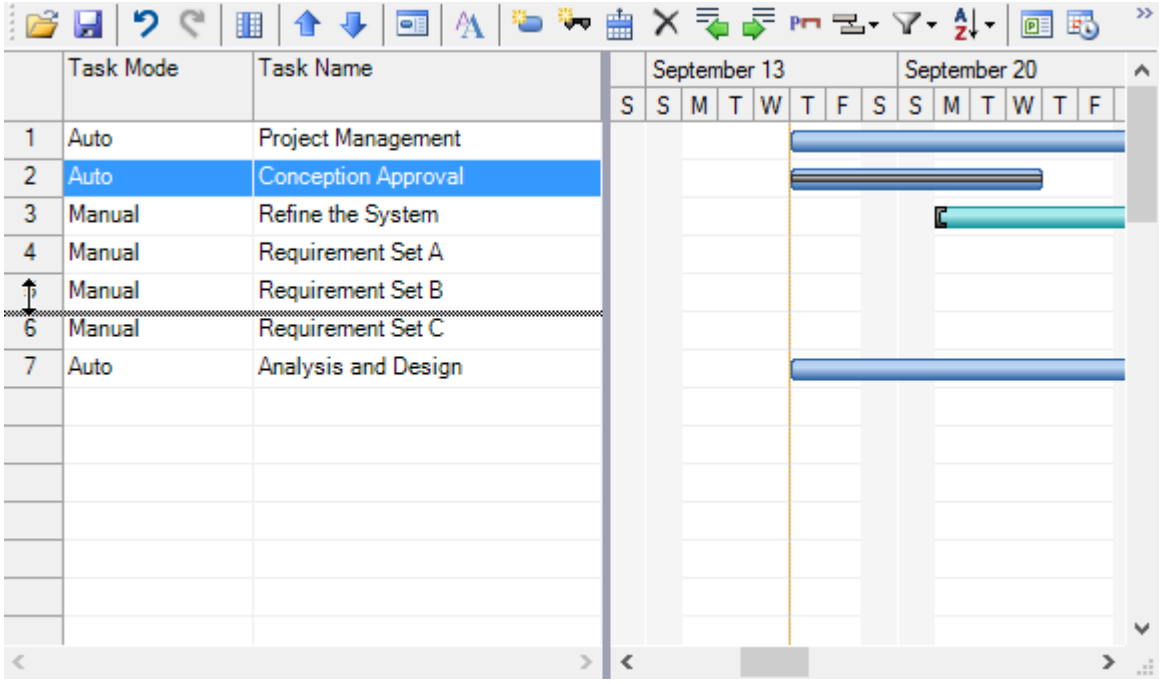
```
private void btnMove_Click(object sender, EventArgs e)
{
    TaskCollection tasks = ganttView.Tasks;
    int index = tasks.IndexOf("New Task");
    if (index > 0)
    {
        Task t = tasks[index];
        tasks.RemoveAt(index);
        tasks.Insert(0, index - 1);
    }
}
```

Drag and Drop Tasks

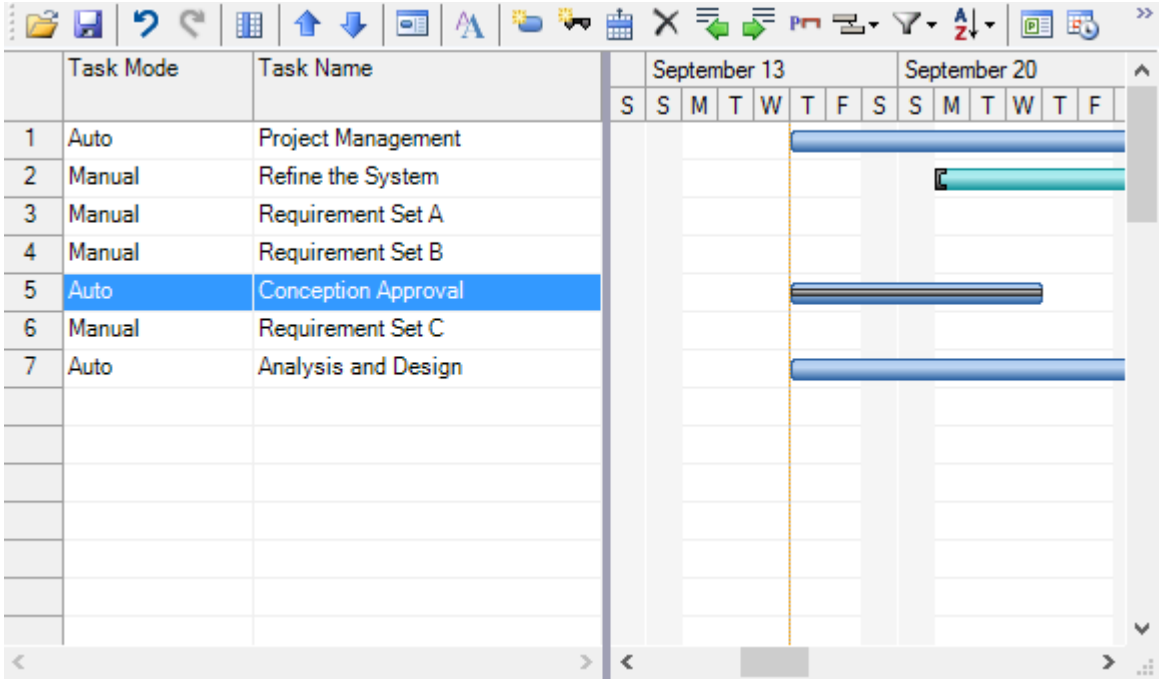
You can now drag and drop a task to change the position of a row within tasks grid. It is also possible to select multiple rows and drag them. To drag and drop a task, perform the steps given below:

1. Select the task you wish to change its row position.
2. Drag the selected task to the desired position where you wish to drop it.
   As you can see in the given image, Conception Approval task is dragged to the row position above Requirement Set C task.

Copyright © 2017 GrapeCity, inc. All rights reserved.



3. Drop the selected task above or below another task to reorder the row positions of tasks.
   In our case, Conception Approval task is dropped at fifth row position, as shown in the image:



Splitting Tasks

GanttView for WinForms allows you to split tasks into several parts to determine times of work and suspensions. You can move the separated parts of a task, increase and decrease period of work/suspension on chart view using mouse. You can also split the selected task from the context menu by clicking Split task menu item. If you need to put a task on hold in the middle, you can easily split the task so that a part of it can be started later. You can also split a task as many times as you need.

Split task feature can be used in situations where a resource already working on a task goes on a leave or is assigned another task on high priority. In the latter case, the resource might need to put the currently assigned task on hold to

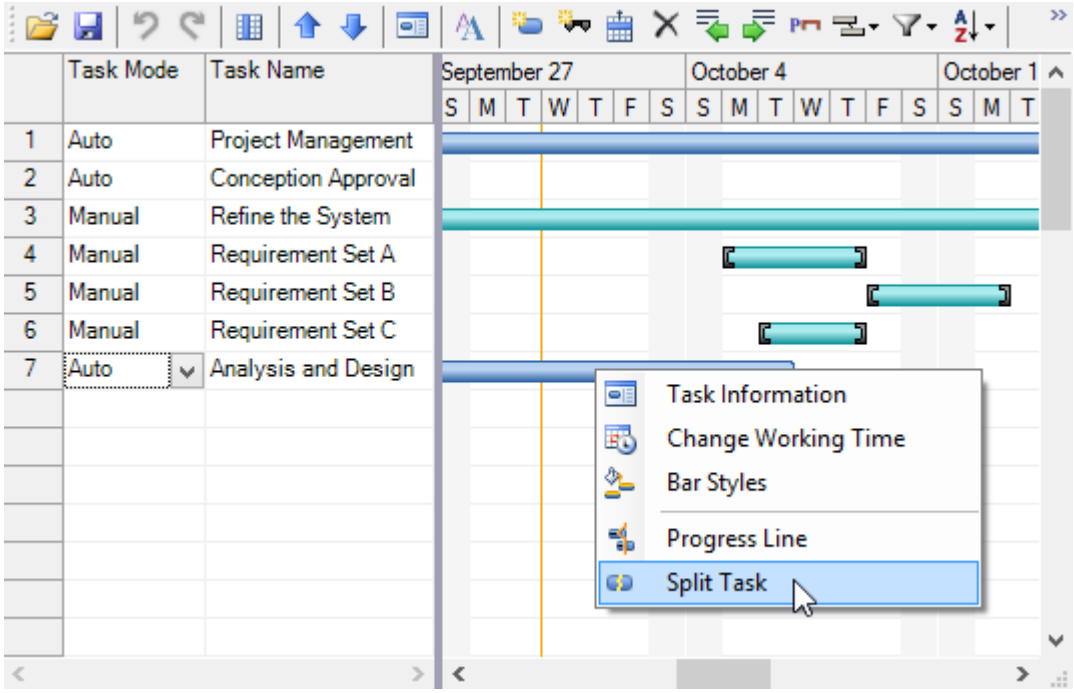
Copyright © 2017 GrapeCity, inc. All rights reserved.

start the newly assigned task. In this situation, the ongoing task can be split into two parts so that the newly assigned high priority task can be started. The resource can resume the interrupted task again as soon as the high priority task gets completed.

Split a task at run time

To understand how to split a task in C1GanttView, you need to follow the given steps at runtime:

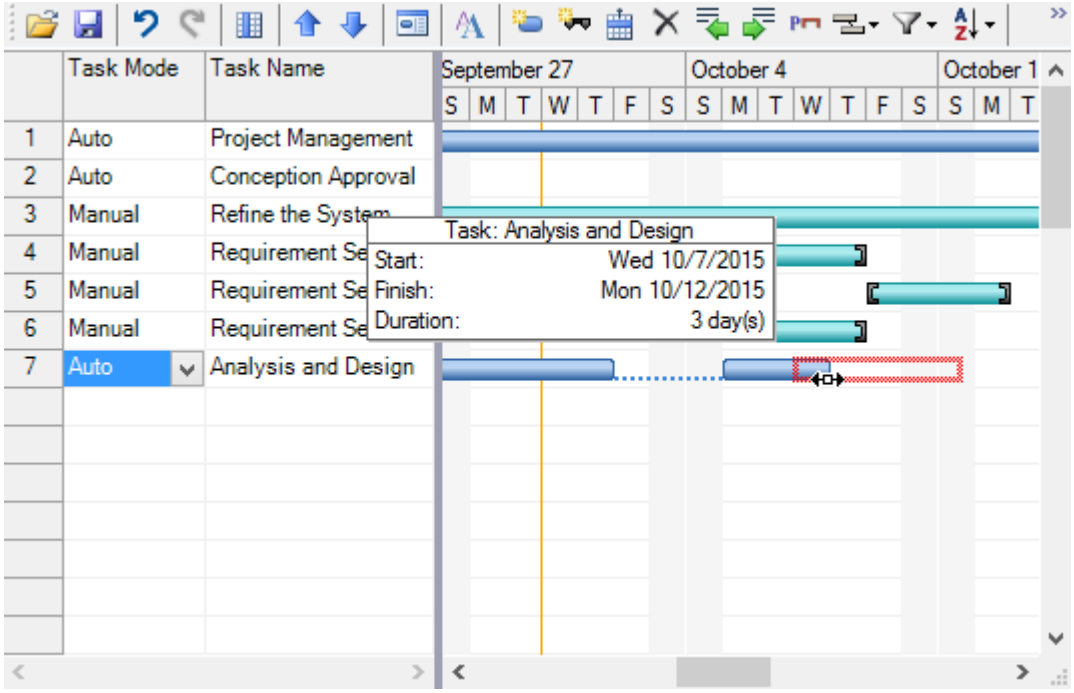
1. Right-click on a task's Gantt bar, on the date where you want the split.
   A context menu will appear.
2. Select Split Task from the context menu as shown in the following image:



   Task will split on the selected date.

3. Drag the second part of the bar to the date from where you want to resume that task again as shown in the image below:

Copyright © 2017 GrapeCity, inc. All rights reserved.



Split a task programmatically

You can also split a task programmatically. To do so, use the following code:

- **Visual Basic**

```
C1GanttView1.Tasks(6).SplitTask(C1GanttView1.Tasks(6).Start.Value.AddDays(15), 3)
```

- **C#**

```
c1GanttView1.Tasks[6].SplitTask(c1GanttView1.Tasks[6].Start.Value.AddDays(15), 3);
```

Inactivate a Task

GanttView control allows you to disable a task by inactivating it. Inactivating disables a task while maintaining its record in the project. When you inactivate a task, it remains in the project plan but does not affect other tasks schedule, project schedule, and resource availability.

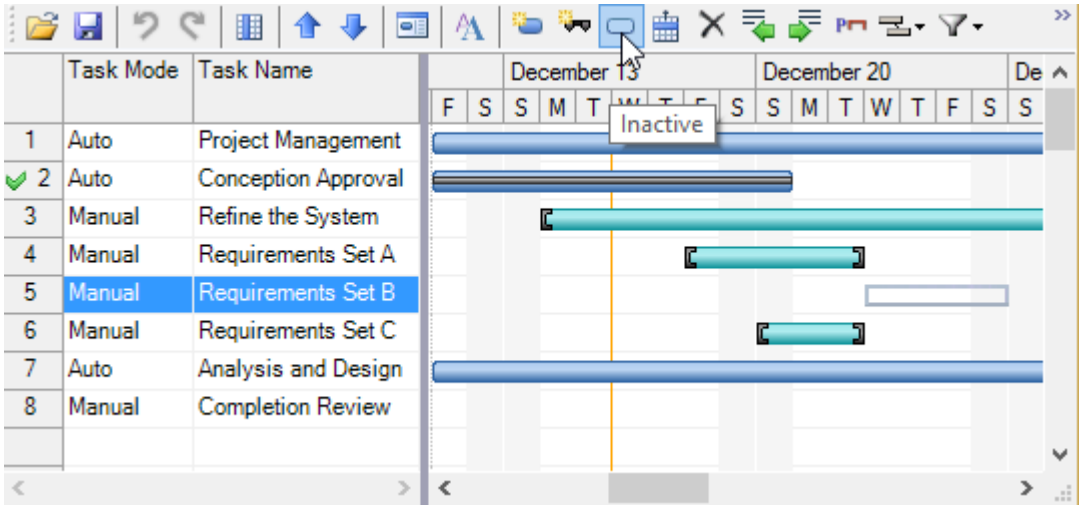
Inactivating tasks can help in building project budgets by disabling the tasks that are no longer required, without deleting them permanently. Deleting a task can affect the schedule of the entire project plan. Hence, inactivating a task is always a better option as you can keep the record of cancelled tasks and reactivate them anytime during the scope of the project.

Inactivate a task at run time

You can inactivate a task by clicking **Inactive** button on the GanttView toolbar. To inactivate a task at run time, follow the steps given below:

1. Select the task you wish to inactivate in the grid. In our case, we have selected **Requirement Set B** task.
2. Click **Inactive** button on the **C1GanttView** toolbar to inactivate the selected task, as shown in the image given below:

Copyright © 2017 GrapeCity, inc. All rights reserved.



As you can see, **Requirement Set B** task in the GanttView control is inactive now.

Inactivate a task programmatically

Inactive property is used to inactivate a task in the grid. The code given below illustrates the use of Inactive property:

- **Visual Basic**

```
C1GanttView1.Tasks(4).Inactive = True
```

- **C#**

```
c1GanttView1.Tasks[4].Inactive = true;
```

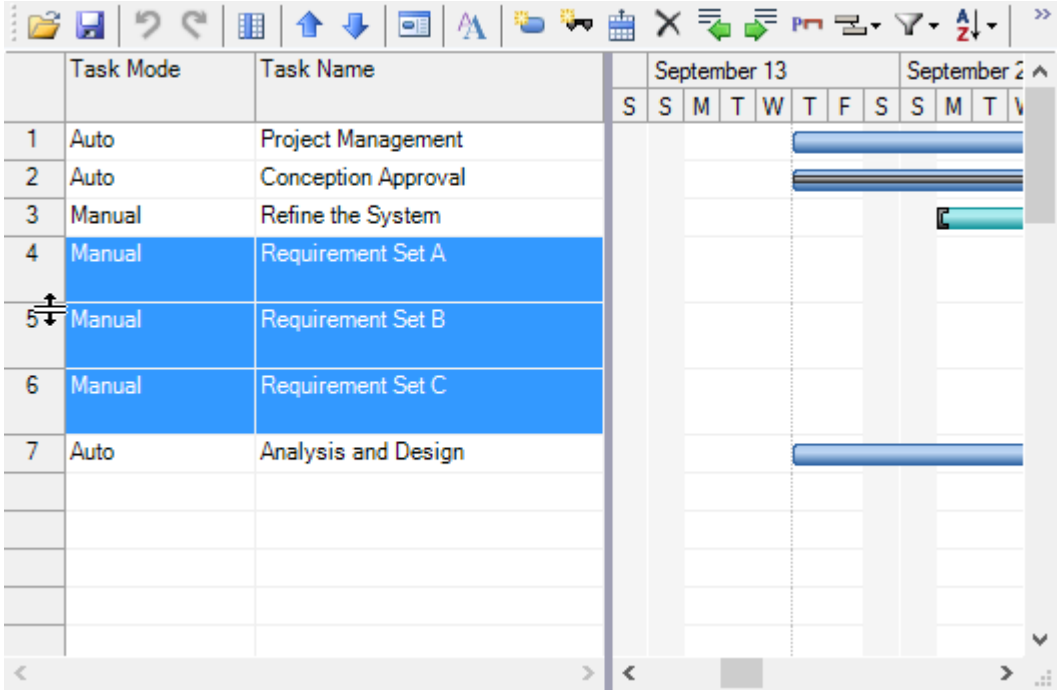
Customizing Multiple Rows and Columns

Rows and columns in GanttView control can easily be customized now. You can select multiple rows or columns and customize them according to your needs. You can indent and outdent the selected tasks, delete the tasks, and change their field styles. You can also select several cells to change field styles for the selected tasks/fields.

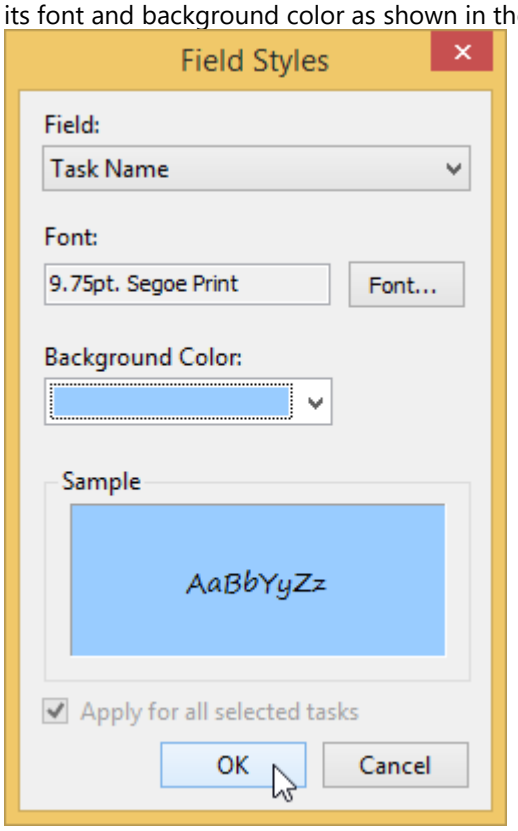
If you need to highlight particular tasks, you can change the font style and background color for that particular task name. You can also change the height or width of the row/column. Follow the given steps to customize multiple rows/columns:

1. Select the rows/columns you want to customize. In our case, we have selected three rows to change their height.
2. Place the cursor at the bottom of a row and drag it to change the height of the selected rows as shown in the image below:

Copyright © 2017 GrapeCity, inc. All rights reserved.

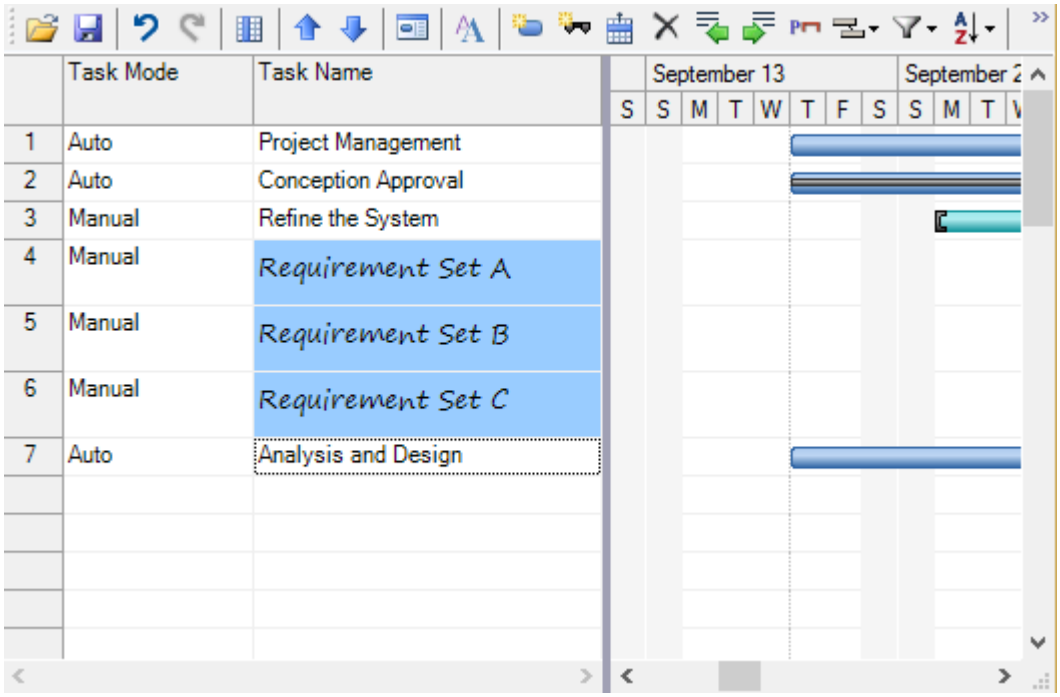


3. Select the fields you want to highlight and select **Field Styles** on the GanttView toolbar.
   Field Styles dialog box will appear.
4. Select the field you want to customize from the drop down. In our case, we selected **Task Name** field, changed its font and background color as shown in the image below:



5. Click OK.
   The customized fields will look similar to the following:

Copyright © 2017 GrapeCity, inc. All rights reserved.



Creating a Milestone

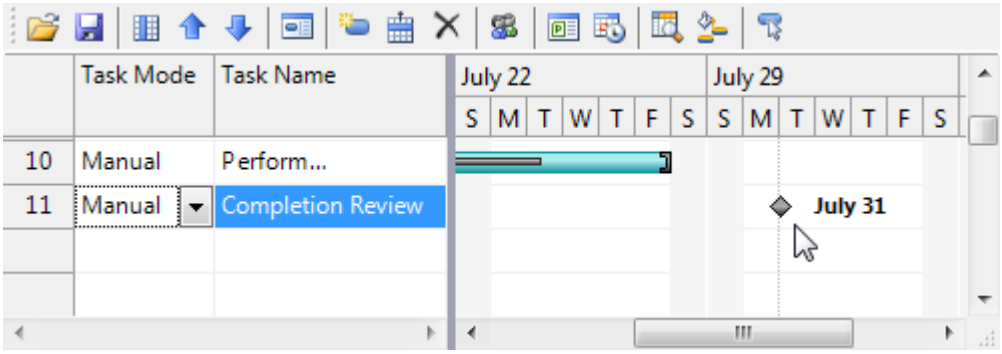
This topic shows how to create a milestone at design time.

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.



Copyright © 2017 GrapeCity, inc. All rights reserved.

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 Programmatically

To programmatically add a predecessor, complete the following:

To write code in Visual Basic

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)

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

To write code in C#

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
```

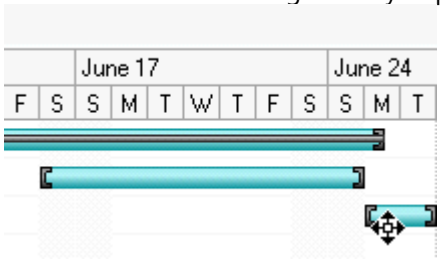
Copyright © 2017 GrapeCity, inc. All rights reserved.

```
        task2.Mode = TaskMode.Manual;
    }
}
```

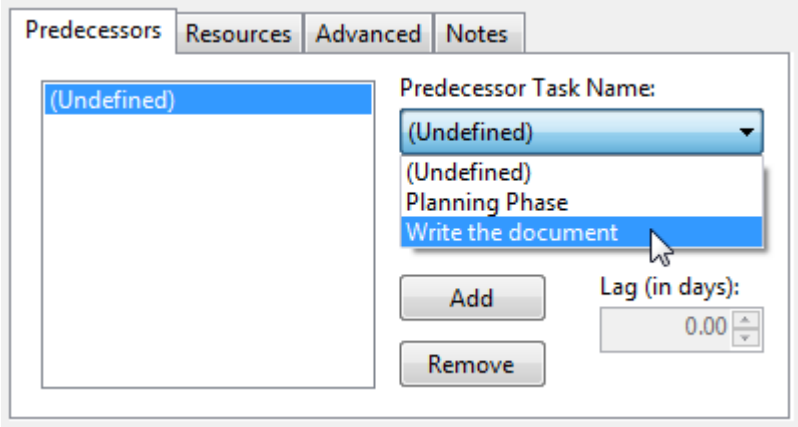
Creating a Finish to Start Predecessor Type

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.



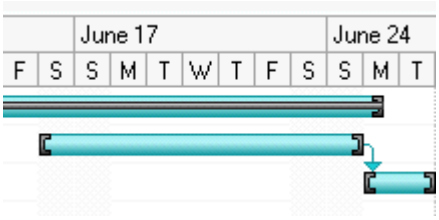
   The Task Information dialog box appears once you double-click the desired task.
2. Select the **Predecessors** tab and click the **Add** button.
3. Click the dropdown arrow from the **Predecessor Task Name**: and select the predecessor task, for example, **Write the document**.



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**.

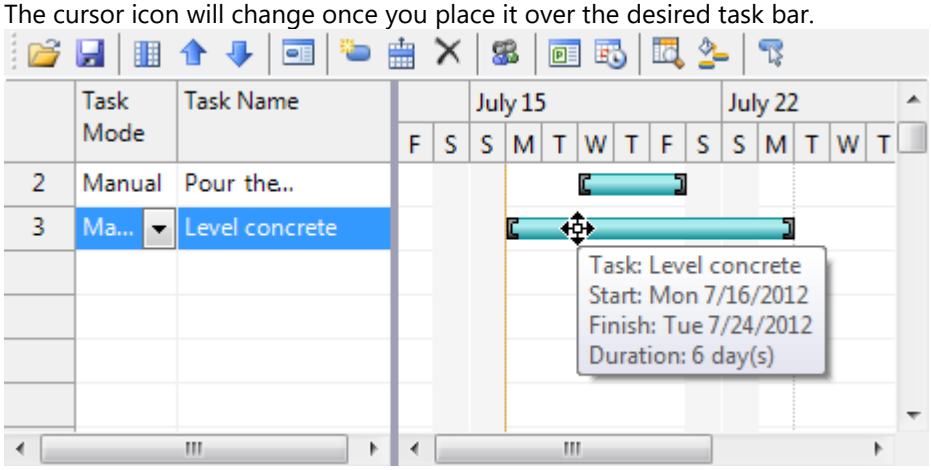


Creating a Start to Start Predecessor Type

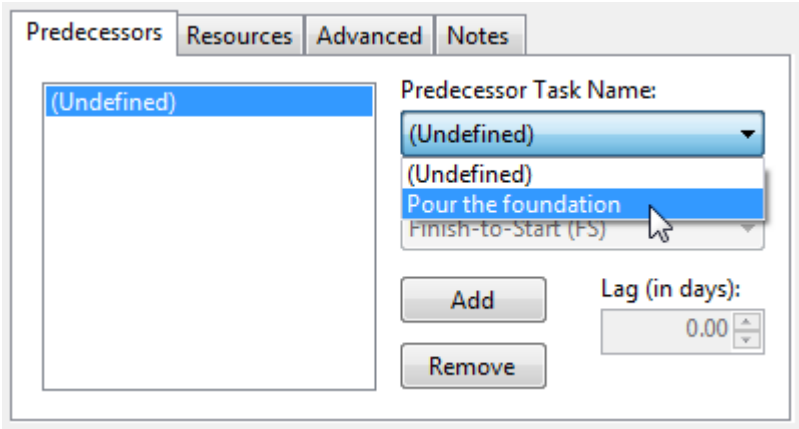
Copyright © 2017 GrapeCity, inc. All rights reserved.

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.



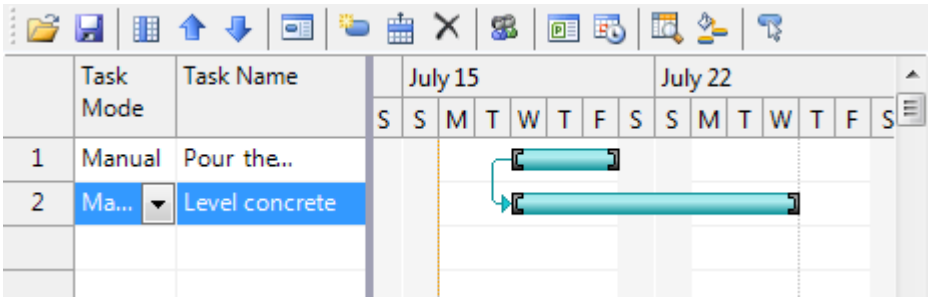
   The Task Information dialog box appears once you double-click the desired task.
2. Select the **Predecessors** tab and click the **Add** button.
3. Click the drop-down arrow from the **Predecessor Task Name**: and select the predecessor task, for example, **Pour the foundation**.



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.

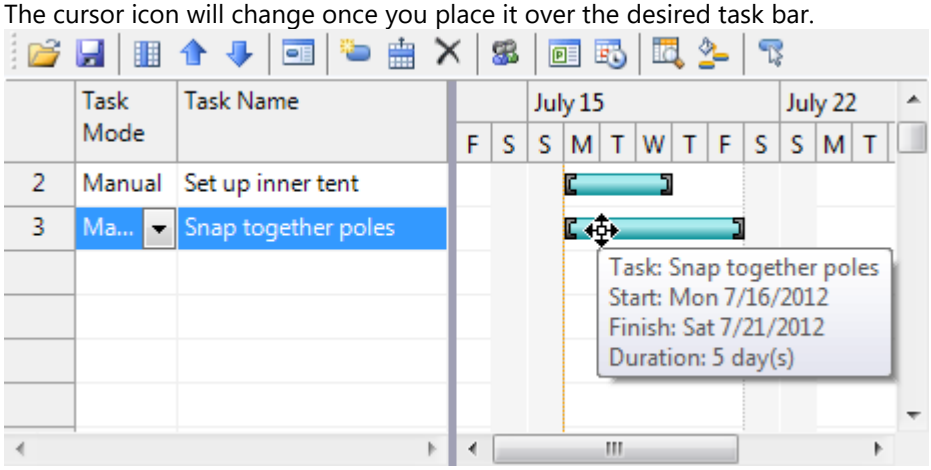


Copyright © 2017 GrapeCity, inc. All rights reserved.

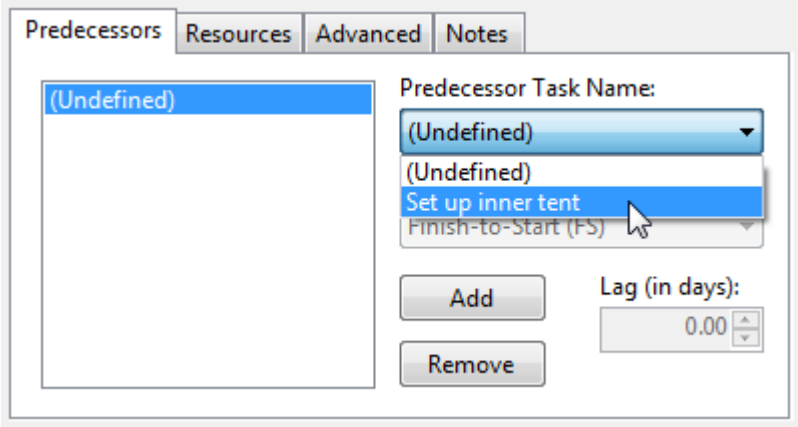
Creating a Finish to Finish Predecessor Type

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 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**.

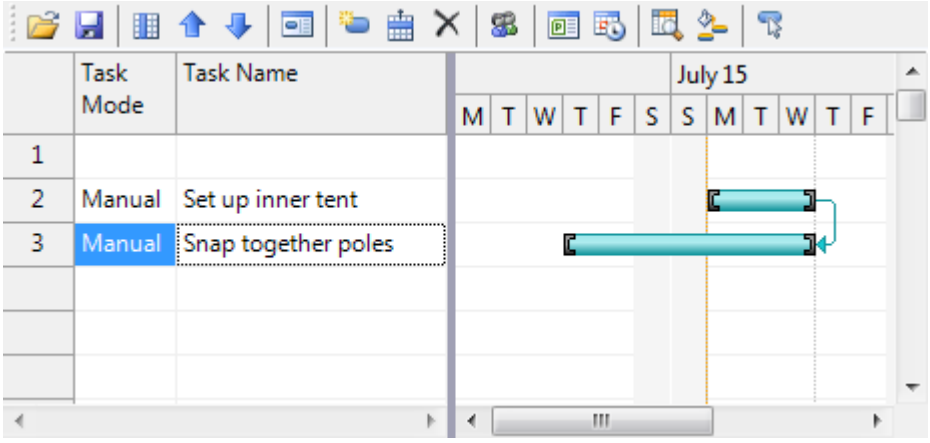


4. Select **Finish-to-Finish (FS)** from the **Predecessor Type**.
   This predecessor type implies that the dependent task, **Snap together poles**, can't finish until the ,**Set up inner tent**, task finishes.
5. Click **OK** to apply the changes and close the **Task Information** dialog box.

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.

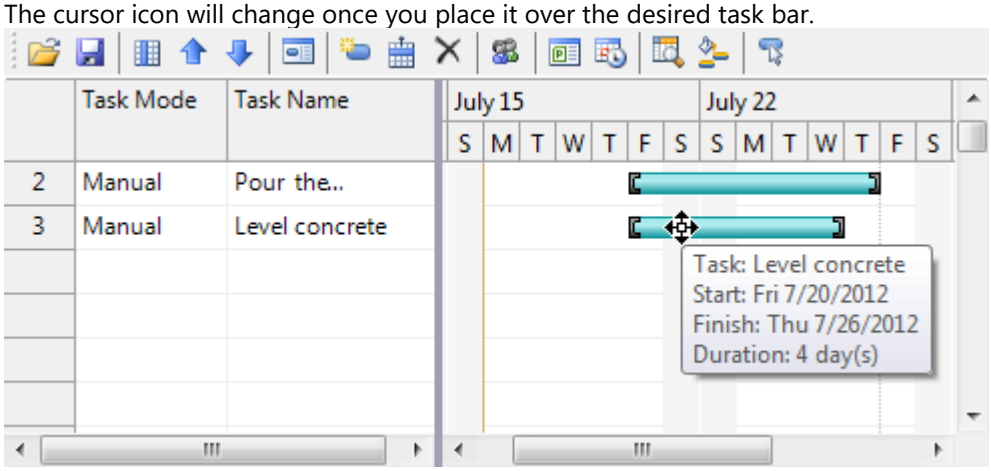
Copyright © 2017 GrapeCity, inc. All rights reserved.



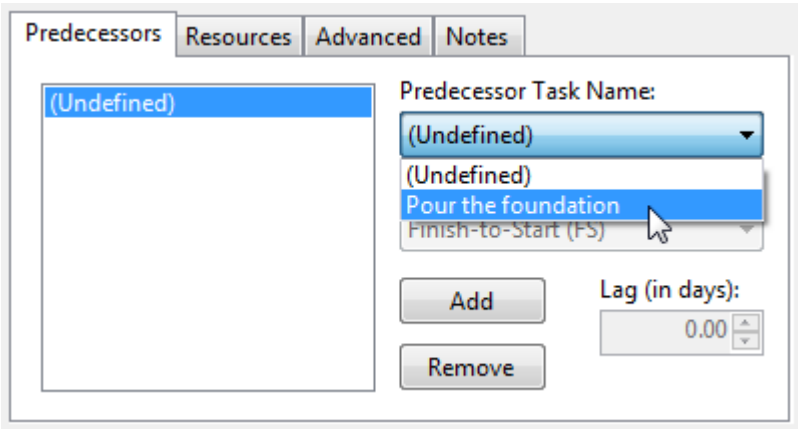
Creating a Start to Finish Predecessor Type

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 dialog box appears once you double-click the desired task.
2. Select the **Predecessors** tab and click the **Add** button.
3. Click the dropdown arrows from the **Predecessor Task Name**: and select the predecessor task, for example, **Pour the foundation**.



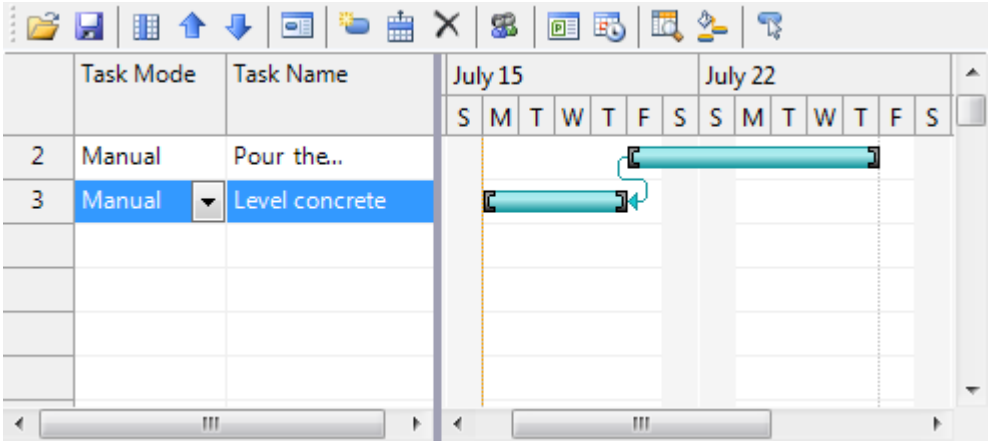
4. Select **Start-to-finish (SF)** from the **Predecessor Type**.
   This predecessor type implies that the dependent task, **Level the concrete**, cannot finish until the start of, **Pour the foundation**.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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. **OK** to save and close the **CustomCalendar.CalendarExceptions Collection Editor**.
10. 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

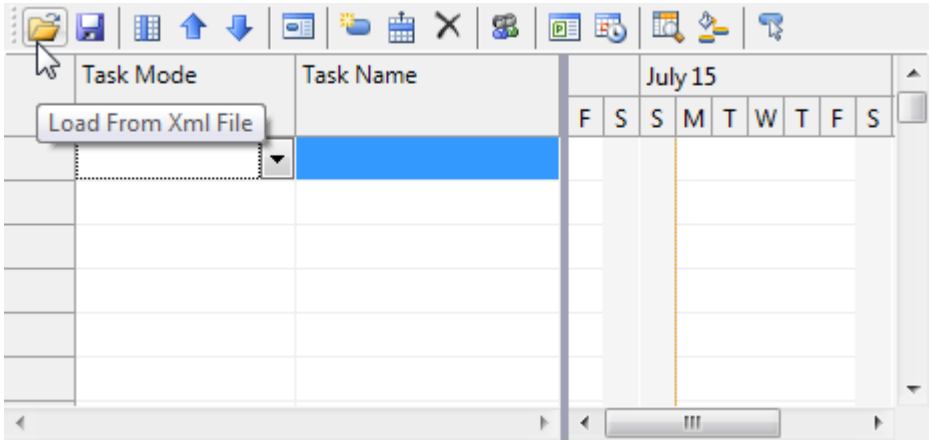
This task shows how to load the C1GanttView as an XML File at run time and in code.

Load C1GanttView as an XML file at run time

Copyright © 2017 GrapeCity, inc. All rights reserved.

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:

To write code in Visual Basic

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
```

To write code in C#

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";
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            try
            {
                ganttView.LoadXml(dlg.FileName);
            }
            catch
            {
                MessageBox.Show("Bad C1GanttView XML.", dlg.Title);
            }
        }
    }
}
```

In addition, you can load MS Project XML file in GanttView by using the ImportFromMsProjectXml method.

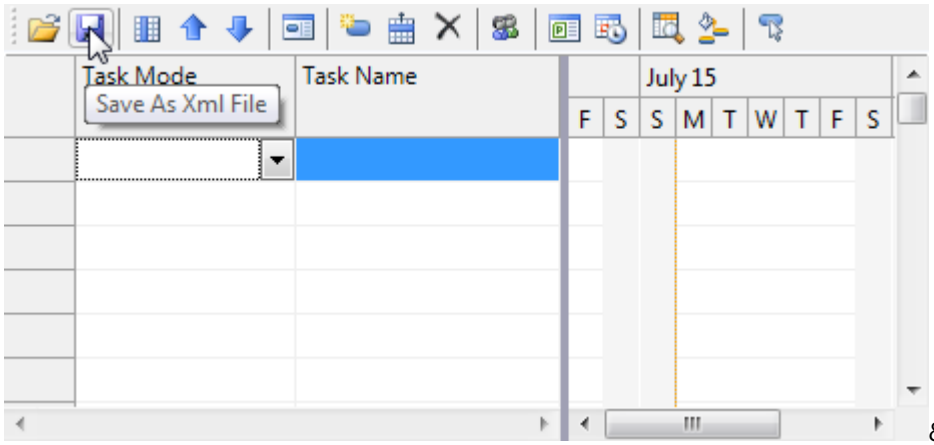
Saving GanttView as an XML File

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:

To write code in Visual Basic

Visual Basic

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

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
            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
```

To write code in C#

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);
        }
    }
}
```

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**.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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:

To write code in Visual Basic

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
```

To write code in C#

C#

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

Change the style of a specific task in code

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

To write code in Visual Basic

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
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
            bs.EndShape = 19
            task3.BarStyles.Add(bs)
        End If
End Sub
```

To write code in C#

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);
    }
}
```

Creating a Custom Column

This topic shows how to programmatically create a custom column.

To write code in Visual Basic

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
```

To write code in C#

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";
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

```
    cc.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
    cc.Width = 65;
    ganttView.Columns.Add(cc);
}
```

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.

To write code in Visual Basic

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
```

To write code in C#

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;
    }
}
```

Changing the Month the Fiscal Year Starts On

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

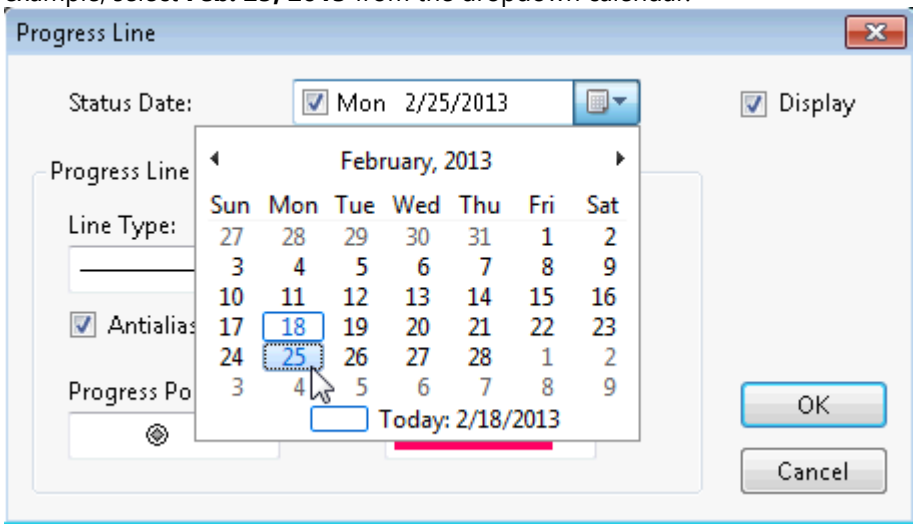
1. Click the **Project Information** button from the **C1GanttView** toolbar button. The **Project Information** dialog box appears.
2. Select the **Calendar Options** tab.
3. Select **April** from the **Fiscal Year Starts in:** dropdown listbox.

Copyright © 2017 GrapeCity, inc. All rights reserved.

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 **C1GanttView** 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.

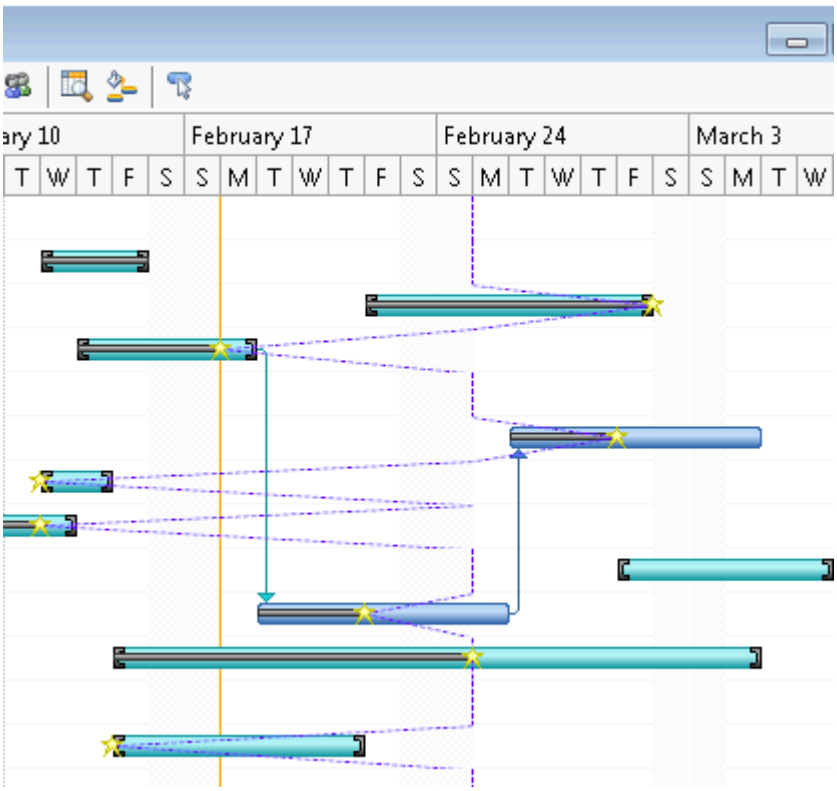


3. Click on the dropdown arrow next to **Line Type** to specify the line type, for example select the dotted lines.
4. Click on the dropdown arrow next to the **Line Color** to specify a new line color for the progress lines. Select a color from the palette, for example, **Purple**.
5. Click on the dropdown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

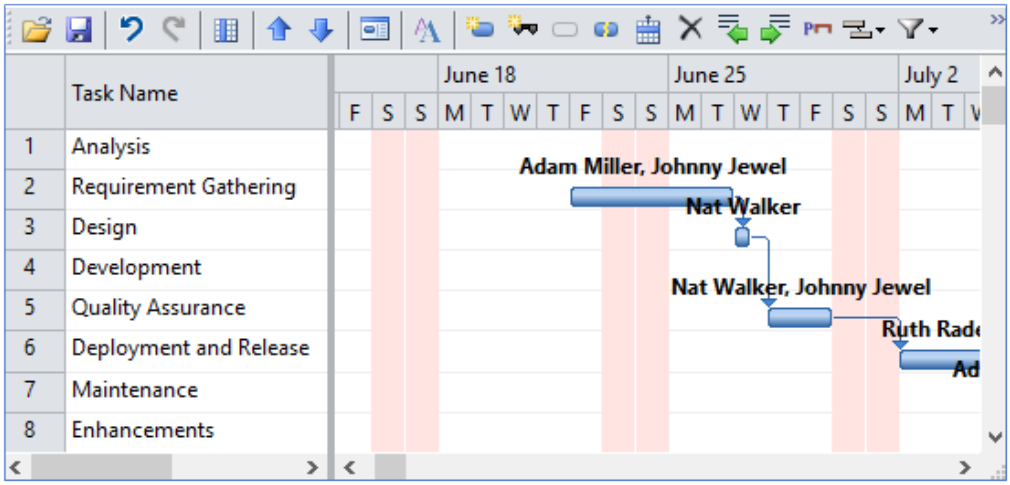
Copyright © 2017 GrapeCity, inc. All rights reserved.



Setting Back Color of Day

GanttView enables users to highlight a specific day of week by setting its back color. For instance, the weekends in a gantt view can be highlighted in a different color to distinguish them from workweek. This can be achieved in code by subscribing the PaintDay event and setting the BackColor property to color specific days.

The following image shows a GanttView with weekends highlighted by a different color.



The following code illustrates how to set back color of weekend.

Visual Basic

```
Private Sub gv_PaintDay(sender As Object, e As PaintDayEventArgs)
        If e.[Date].DayOfWeek = DayOfWeek.Saturday
        OrElse e.[Date].DayOfWeek = DayOfWeek.Sunday Then
            e.BackColor = Color.MistyRose
        End If
    End Sub
```

Copyright © 2017 GrapeCity, inc. All rights reserved.

C#

```
private void gv_PaintDay(object sender, PaintDayEventArgs e)
        {
            if (e.Date.DayOfWeek == DayOfWeek.Saturday || e.Date.DayOfWeek == DayOfWeek.Sunday)
            {
                e.BackColor = Color.MistyRose;
            }
        }
```

Copyright © 2017 GrapeCity, inc. All rights reserved.