

What is a Gantt chart?

Article Index

What is a Gantt chart?

Dependency relationships

Page 1 of 2

More than just a chart, a Gantt Chart is a UI used to model tasks with interdependencies to show how a project fits together. A Gantt chart control not only gives you an easy way of constructing a model but can actually do some planning for you.

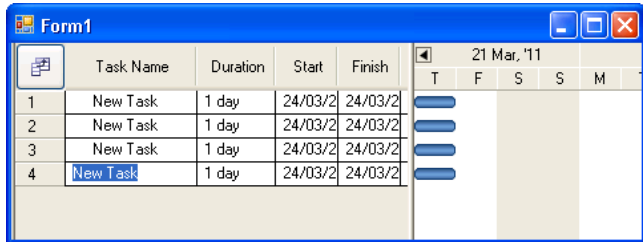
Sometimes you have to ask a simple question to get a simple answer. In most cases what a chart component does is fairly obvious but sometimes they can be a bit more mysterious. Even looking at examples only clarifies how the control can be used in specific cases not the general principles.

So it is with the Gantt chart. It has an odd name and when you look at a typical example it can be difficult to see what is going on. In practice the Gantt chart isn't complicated and once you know what it is all about a range of non-standard applications open up.

In this article we look at the Gantt chart in general and to make things concrete we will use the Infragistics Gantt chart component. After you have seen how it all works you can contemplate creating your own component but you will also be in a good position to estimate how difficult the task might be,

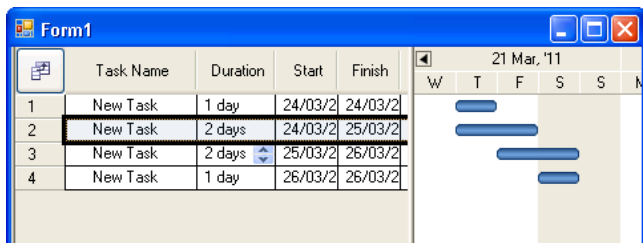
Download a free trial evaluation of the Windows Forms controls from the Infragistics site. Install it and open a copy of Visual Studio and start a new Windows Forms project. Place a single UltragantView control on the form and size the form so that you can see all of the control.

A Gantt chart is simply a plot of start and end times. Each horizontal bar indicates a start date, end date and hence a duration. To add a bar simply run the program, yes you don't need to add any code, and right click on the Gantt control and select Insert task - and few while you are at it.



You can see that each bar can represent a task or activity. By dragging it left or right you change when the task happens without changing its duration. By dragging either the start or end dates you change how long

the task takes.



A set of horizontal bars can be considered to make up a project consisting of tasks.

So that's it. A Gantt chart is a set of bars indicating the timing and duration of tasks

that make up a project.

If that is all you want it to be, yes, this is what a Gantt chart is - but it can also have some extra facilities that help you model the way that tasks relate to one another.

Consider for a moment a task. In most cases it will be composed of other sub-tasks which have to be completed in some sort of order before the entire task is completed. You can use a Gantt chart to model this.

Start with a single task and right click on the Gantt control. Select Add Sub task and do this twice. Now you have a task that is composed of two sub-tasks and you can see that the way that the top level task is drawn has changed.

	Task Name	Duration	Start	Finish
1	NewTask	1 day	26/03/	26/03/
2	New sub-ta	1 day	26/03/2	26/03/2
3	New sub-ta	1 day	26/03/2	26/03/2

In the default form the subtasks appear to occupy the same time slot and they are assumed to run in parallel. However if the first subtask has to be complete before the second then we have a more interesting

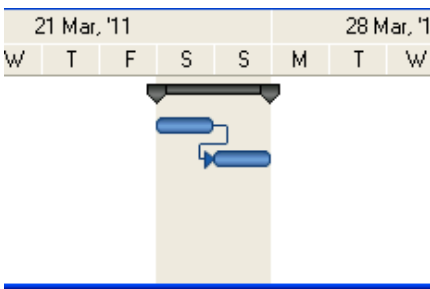
model.

To set this up right click on the second task and select Task Information, click on the Predecessors tab and enter the row number of the task that has to be completed before this task starts - row 2 in this example.

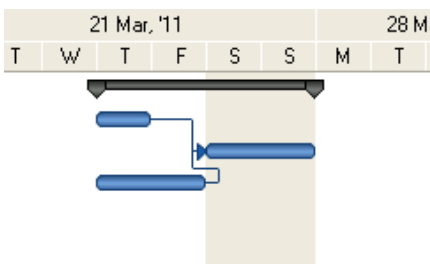
Row Number	Task Name	Task Dependency Type
2	New sub-task	Finish-to-Start (FS)

You also need to select Finish to Start or FS relationship. If you do this you will see the new Gantt chart now shows a task that takes as long as the two sub-tasks task to

complete one after another - the Gantt chart has already worked something out from the model you are building.



This is a general principle. If you add multiple subtasks and link them together so that the start time of a given subtask depends on the finish time of a set of subtasks then that task will only start when the final subtask is complete. For example:



In this case the first subtask is complete after one day but the third subtask now takes two days and so the second subtask cannot start until then. Notice that once again this is worked out for you by the Gantt control. However if you enter data into any of the fields

then your entered values take precedence. To see the natural computed values for a field simply clear it.

Article Index

What is a Gantt chart?

Dependency relationships

Page 2 of 2

Dependencies

Now that you have seen the way that basic Finish to Start FS relationship between tasks you can probably guess that there are others. Tasks can depend on each other in a number of ways and you can select these in the Task Dependency Type field:

- Finish to Start - the task has to finish before this task can start
- Start to Start - both tasks have to start at the same time
- Finish to Finish - both tasks have to finish at the same time
- Start to finish - an inverse FS link the predecessor can't start until the current task finishes.

You can see that by creating tasks and linking them together with task dependencies you can build a model of how the tasks determine the overall behaviour of the project. You can take it further by assigning resources to tasks and constraints such as a particular task has to be completed by a particular date and so on.

The point is that the Gantt chart isn't just a chart. It is the visible surface of a project model. Because the user can interact with the UI it presents it is more than just a static chart because it provides a way for the user to create and interact with the model.

The programmer's role

At this point I hope you are impressed by this simple component. It has the power to keep your projects - whatever they are - on target or at least let you know very precisely why they are not! However as a programmer you might be thinking that its all been done for you. Why bother to use a Gantt chart component in your application? Why not simply get a project management application and use a complete solution?

There are a number of very good reasons for preferring a Gantt chart component.

The first is that project management programs are often very complex applications that take a long time to learn even if you only want to do fairly simple things.

Then there is the issue of where the data is generated. If the details of tasks naturally arise within your application then presenting it in Gantt chart form and performing some simple modeling actions makes good sense.

Everything that you have just been doing using the UI you can, of course, do in code. If you have used almost any sophisticated component you will already have a good idea how this is going to be done.

There is a Tasks collection which you can use to add and generally access the task objects that make up the project. So for example, to add a task you could use something like:

```
ultraGanttView1.CalendarInfo.Tasks.Add(  
    DateTime.Today,  
    TimeSpan.FromDays(5),  
    "Task1",  
    "unassigned_project");
```

Tasks themselves have a Tasks collection and this is how you create complex tasks consisting of sub-tasks. For example to create a task composed of two sub-tasks you would use:

```

Task Task1 = new Task { Name = "Task1" };
Task1.Tasks.Add(
    DateTime.Today,
    TimeSpan.FromDays(2),
    "Task2");
Task1.Tasks.Add(
    DateTime.Today,
    TimeSpan.FromDays(2),
    "Task3");
ultraGanttView1.CalendarInfo.Tasks.Add(Task1);

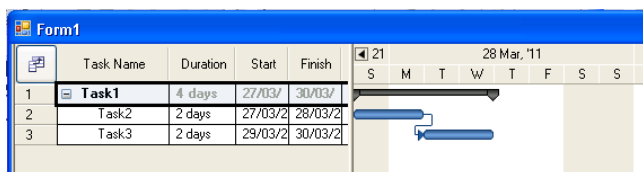
```

You can also model the relationships between tasks using the Dependencies collection. For example to set the two sub-tasks to have a FS dependency you would use something like:

```

Task Task1 = new Task {Name = "Task1"};
Task Task2=Task1.Tasks.Add(
    DateTime.Today,
    TimeSpan.FromDays(2),
    "Task2");
Task Task3=Task1.Tasks.Add(DateTime.Today,
    TimeSpan.FromDays(2), "Task3");
Task3.Dependencies.Add(Task2,
    TaskDependencyType.FinishToStart);
ultraGanttView1.CalendarInfo.Tasks.Add(Task1);

```



You can see how this would continue to build a complex model of relationships between events. If this is all you need then - job, or should that be task, done! But it is

worth pointing out that there are lots of sophisticated processes just waiting to be implemented in code. For example, sometimes tasks end and another cannot start immediately because it also depends on another task. This hiatus is called 'slack'

and it is the slack in the project that is interesting because it sometimes allows tasks to overrun without effecting the entire project. It is also true that slack can be the source of inefficiencies - its where resources might stand idle.

Then there is the best known of all project management concepts - the critical path - i.e the sequence of tasks that does determine the total time the project takes. The critical path is important because any delays to a task that is on the critical path delays the project.

All of these concepts and more are fairly easy to take advantage of in code. You can make use of the Gantt chart and the model that it makes visible for all sorts of data analysis, costings, what-if simulations and so on. Some of these you will find in complete project management packages but now you have the power to tailor them exactly and provide data exchange and sharing with other applications.

Gantt charts are fun - I hope you agree.

Download a free trial evaluation of the Windows Forms controls, including the Gantt chart, from the Infragistics site.