

# Practical Project Management and Tracking using Microsoft Project

by Marvey Mills

## Contents

1	Introduction .....	3
1.1	Useful information .....	3
1.1.1	Using This Guide .....	3
1.1.2	Microsoft Project Versions .....	3
1.1.3	Menu Shorthand .....	4
1.1.4	Copyright and reuse .....	4
2	Starting a new project .....	5
2.1	Default working Time .....	5
2.2	Project Calendar .....	6
2.3	Project Start Date .....	9
3	Adding resources & Availability.....	11
3.1	Project Holidays .....	11
3.2	Adding and setting up resources .....	12
3.3	Groups .....	14
3.4	Personnel Holidays .....	14
3.5	Block availability .....	16
4	Adding Tasks .....	18
4.1	Enter task blocks .....	18
4.2	Entering basic tasks.....	19
4.3	Enter explicit dependencies as constraints .....	20
4.4	Assign Resources.....	22
4.5	Level the Plan .....	25
4.6	Arrange Tasks Using Priority .....	28
4.7	Enter Milestones .....	30
5	Other Scheduling and Planning .....	34
5.1	External Constraints and Milestones .....	34
5.2	Baselining .....	38
6	Tracking Against a Plan .....	40
6.1	Tracking by Percentage Complete .....	40
6.2	Tracking by Work.....	41
6.3	Task Start Dates .....	42
6.4	Putting Tracking into Practise .....	43
6.4.1	Gathering and Recording Actuals .....	43
6.4.2	Moving Unfinished Tasks .....	43
6.4.3	Levelling the Plan .....	46
6.4.4	The Tracking Gantt .....	47

## 1 Introduction

I have been working in the field of Project Management of systems and software delivery for many years and for many employers. A common theme running through most of that experience has been the use of Microsoft Project (MS-P) to plan the project before it starts and to track progress against it during the project.

Most project managers I have met in that time that use MS-P have a love-hate relationship with the application. At times and in certain circumstances it seems to provide exactly what a project manager needs to do their job. At other times it seems to conspire against us almost willfully to make ostensibly the simplest task impossible to complete.

I have seen and experienced myself vital project plans, that took many hours of work to create and maintain, thrown away in disgust and frustration because the plan got tied into a knot and the poor project manager was unable to unpick the resulting mess.

Long ago I learnt some practical lessons and easily repeatable techniques on how to use Microsoft Project to achieve what I needed as a project manager, whilst avoiding the pitfalls and elephant traps that the application sets for the unwary user. Since then I seem to have spent many hours and days passing this information on to other project managers in every organization I have worked for.

Finally I took the obvious step and wrote it down in order to make my life easier in the future. I do not claim to be an expert in the use of MS-Project, there are many areas of the product that I don't know how to use. Nor do I claim that these simple techniques are all you need to know to manage software projects effectively, are some kind of "best practice" or represent the pinnacle of Microsoft Project user guide perfection. All I claim is that these techniques have been honed in the course of real software project management and work for me; I use them all every day in my real job. I am passing them on in the hope they may provide some small comfort to a frustrated fellow project manager tearing their hair out because Microsoft Project has mangled their latest work of creative genius!

### 1.1 Useful information

#### 1.1.1 Using This Guide

This document is designed to be read in a linear fashion alongside setting up a new project in MS-P. It starts with what to do as soon as you have created the new project and ends with tracking progress against that plan. Microsoft Project has an incredible amount of features and functionality and furthermore allows the project manager to use any of it they want to, mostly in any order they desire. Mis-application of tools and features at the wrong time is one of the key ingredients of a mangled project plan. Following this guide in the order the tasks are presented from the moment you first create the new Project will help you to avoid "death by MS-Project"!

The guide does not purport to be a beginners' guide to MS-Project. It assumes the reader is familiar with the application, how to navigate around it and how to use elements of the host operating system to undertake basic file handling and other menial tasks.

#### 1.1.2 Microsoft Project Versions

The version used in the images and diagrams in this guide was Microsoft Project 2003 – Standard Edition. I believe, but I cannot guarantee, that the techniques and procedures described here work fine on older and later versions of Project. Certainly I have used all versions of MS-Project up to 2003 and employed these techniques. However I have no personal experience of later versions nor Project Server so cannot comment on the efficacy of this document in respect of those.

### **1.1.3 Menu Shorthand**

Text in italics, such as '*Tools => Change Working Time => "Options" Button*' describes the path the user takes from the top level to find the screen being referenced in the text. In this example a more verbose wording might read: Select the Tools menu, select Change Working Time from that menu, then press the Options button.

### **1.1.4 Copyright and reuse**

This document is copyright Marvey Mills and may be reproduced by any means as long as:

- There is no charge to the recipient or user.
- Any reuse of any portion of the document must credit Marvey Mills

## 2 Starting a new project

### 2.1 Default working Time

Tools => Change Working Time => "Options" Button

The screenshot shows the 'Options' dialog box with the 'Calendar' tab selected. The title bar reads 'Options' and the tab is labeled 'Calendar'. The main area is titled 'Calendar options for 'Project1''. It contains several settings:

- Week starts on: Monday (dropdown)
- Fiscal year starts in: January (dropdown)
- Use starting year for FY numbering
- Default start time: 09:00 (text box)
- Default end time: 17:15 (text box)
- Hours per day: 7.50 (spin box)
- Hours per week: 37.50 (spin box)
- Days per month: 20 (spin box)

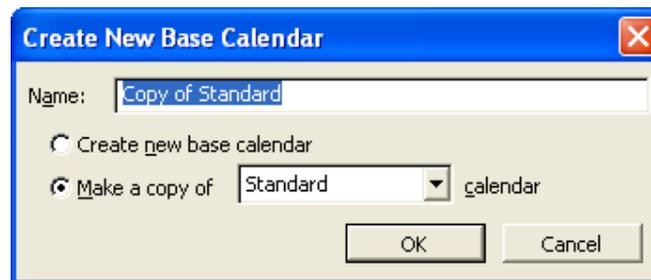
There is a text box on the right side of the dialog that reads: "These times are assigned to tasks when you enter a start or finish date without specifying a time. If you change this setting, consider matching the project calendar using the Change Working Time command on the Tools menu." At the bottom right, there is a 'Set as Default' button. At the bottom left, there is a 'Help' button. At the bottom center, there are 'OK' and 'Cancel' buttons.

Ensure the fields match the numbers above, or your own organisation's working practises. Pressing the "Set as Default" button after completing the fields will means MS-Project will always use your numbers by default and you will not need to go through this step again.

## 2.2 Project Calendar

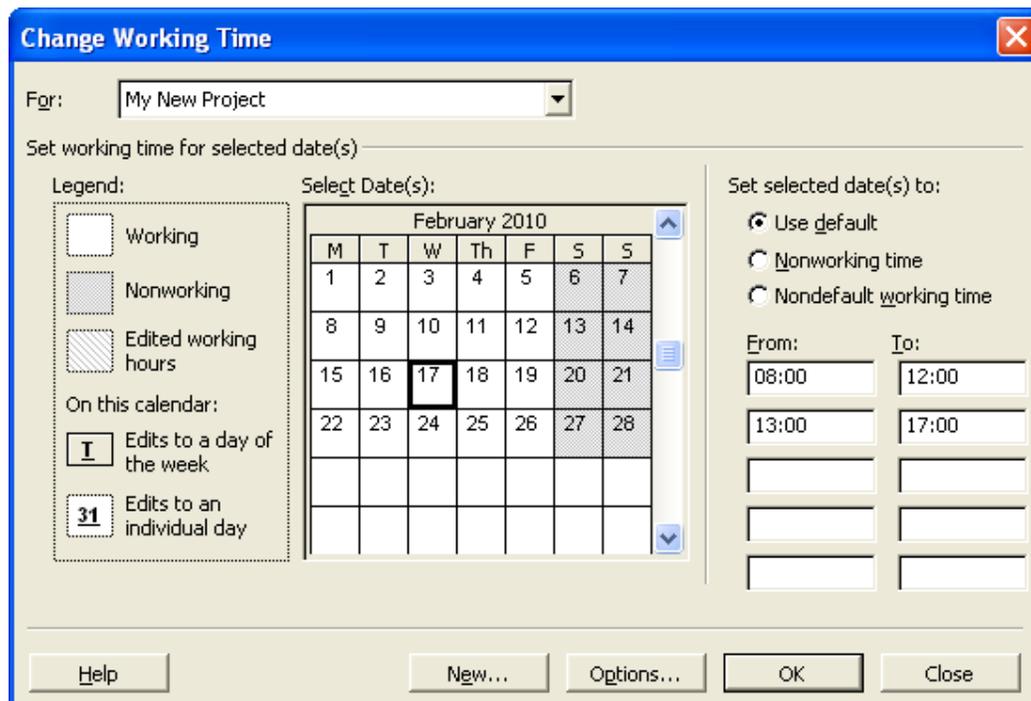
The first thing to do after creating a new project is to create the project calendar. If you do this at any time after resources or tasks have been added then the way MS-Project handles time and days will introduce confusion downstream.

Tools => Change Working Time => "New" button

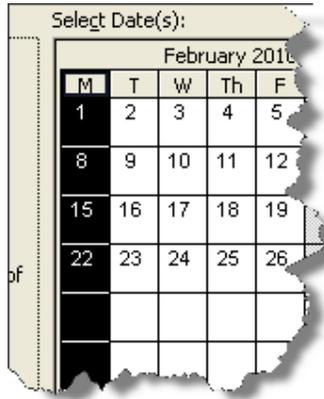


It does not seem to make much difference whether you make a copy of Standard or create a brand new one.

Enter a name for the Calendar using the project name, for example "My New Project" and press ok.

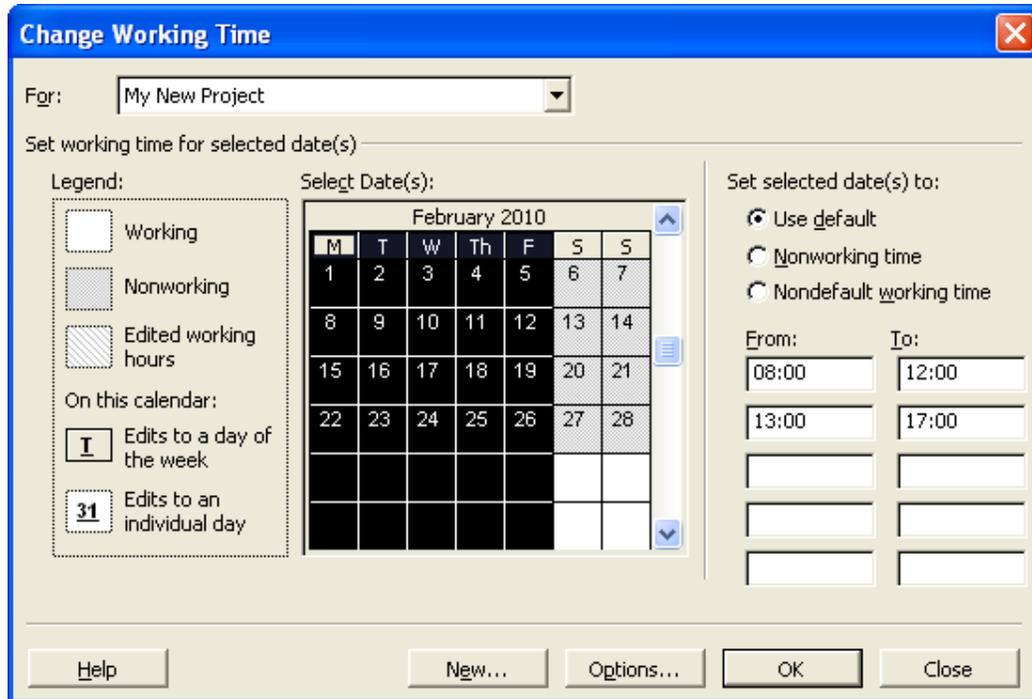


You will notice that the calendar default working time does not match the default working time entered in Section 3.1 above. This seems to be a quirk of MS-Project and I have never been able to make the default times match.

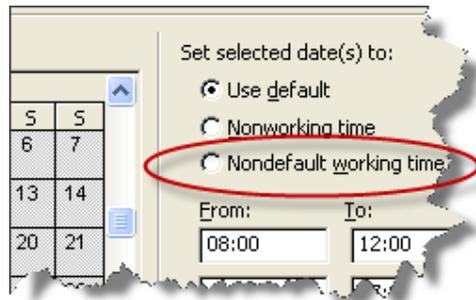


To remedy this you must set the actual working times and the best time to do this is when there are no resources or tasks on the project. The method of setting a new working time across the entire calendar, like so many things in MS-Project, is not very intuitive. First click on the 'M' for Monday. This will highlight all of the Mondays in the current month (it doesn't matter which month you use for this).

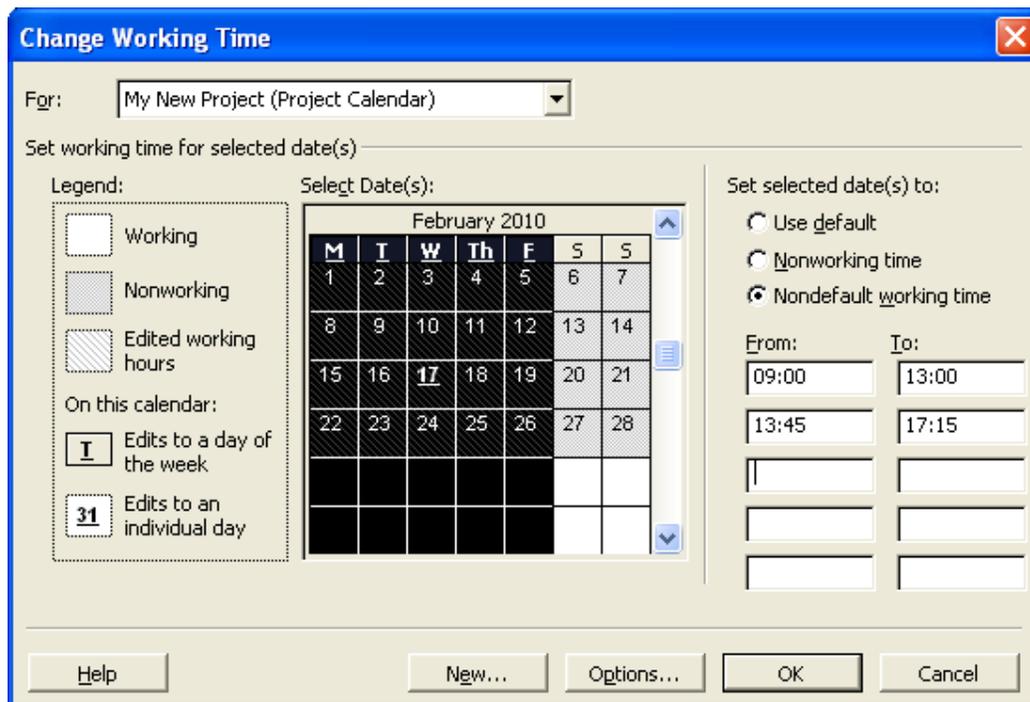
Next drag the cursor, whilst still pressing on the left-hand button, across the days of the week header until you get to 'F' for Friday. Release the left button. Although it is not obvious, you have in fact selected *all* weekdays in the entire calendar at this point!



Click on the 'Nondefault working time' radio button



You may now enter the standard daily working times in the From/To fields and you should enter the same working times you entered in section 2 Starting a new project. Note that you *must* tab away from the last field you enter in order to save it- Navigating away from the screen, by pressing OK, with the cursor still in a field you have changed causes your change to that field to be lost without warning. In fact it is worth going back into the screen after you have finished to ensure that all the correct daily times are stored as modifying these *after* resources and tasks are applied to the project causes mayhem.



It is entirely possible that resources on a project will not actually work the hours entered here- some may start and finish earlier, others later; some may take their lunch break at a different time. This is immaterial unless you are working on a project where the actual time of days that tasks are worked is critical.

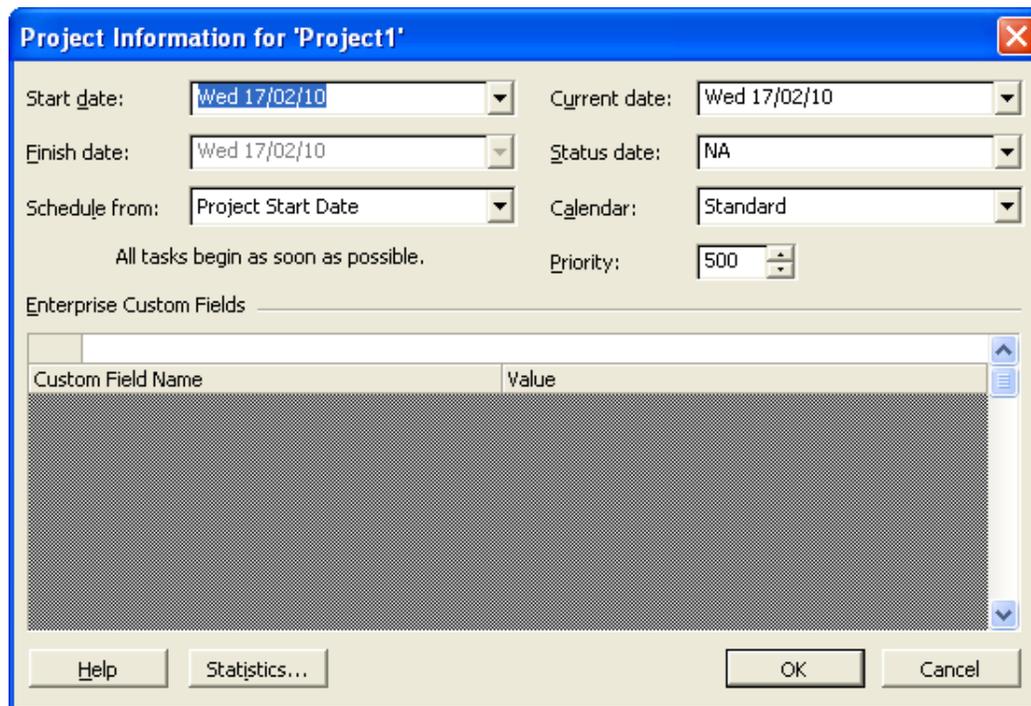
The most important feature of entering these times is that the right number of hours per day is shown as the working time, 7.5 hours is usual though it should match your organisation's working times. This is important because elsewhere in the system a working day is defined at 7.5 hours. This means that if you enter a task with a duration of one day, MS-Project knows that means 7.5 hours work- If you then allocate it to someone who is working, say, 8 hours a day according to this calendar, then MS-Project thinks the task will take them less than a day (i.e. 7.5 hours of their 8-hour day) and begin a new task on the same day. The difference is

minimal but adds up over the course of a project to a significant difference between planned milestone dates and actual achievable milestone dates.

## 2.3 Project Start Date

Next set the project start date and calendar.

*Project => Project Information*



**Project Information for 'Project1'**

Start date: Wed 17/02/10      Current date: Wed 17/02/10

Finish date: Wed 17/02/10      Status date: NA

Schedule from: Project Start Date      Calendar: Standard

All tasks begin as soon as possible.      Priority: 500

Enterprise Custom Fields

Custom Field Name	Value
-------------------	-------

Buttons: Help, Statistics..., OK, Cancel

Ensure the start date is set to the correct date (usually today). It will not matter if this is changed later, though there will be difficulties if you try to make it later and work has already been done on tasks prior to the new start date.

Set the Calendar to be the calendar you set up in 1.2.

**Project Information for 'Project1'**

Start date: Wed 17/02/10      Current date: Wed 17/02/10  
Finish date: Wed 17/02/10      Status date: NA  
Schedule from: Project Start Date      Calendar: My New Project  
All tasks begin as soon as possible.      Priority: 500

Enterprise Custom Fields

Custom Field Name	Value
-------------------	-------

Buttons: Help, Statistics..., OK, Cancel

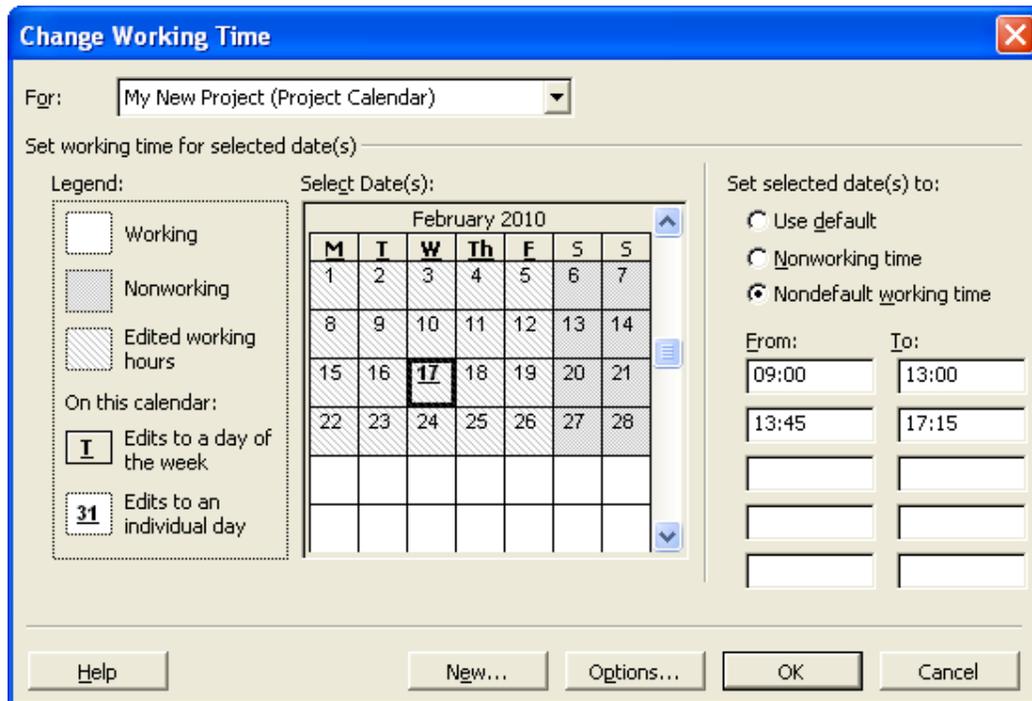
The new project is now ready to enter tasks and resources. Save a copy at this point.

### 3 Adding resources & Availability

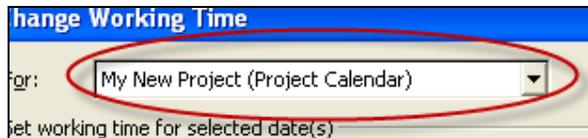
#### 3.1 Project Holidays

MS-Project holds a central calendar for the entire project, plus a calendar for each resource assigned to the project. Use the project calendar to enter project holidays- These will automatically be recognised at an individual resource level. This is where you would enter Bank Holidays and other generic non-working time.

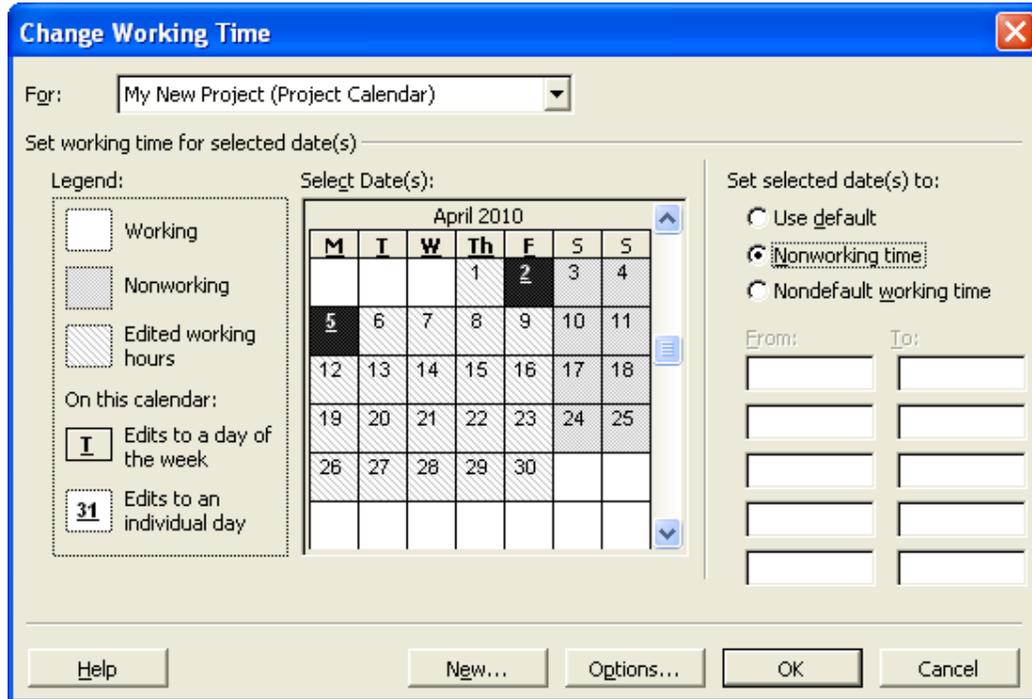
Tools => Change Working Time



Ensure that the Calendar you are modifying is the correct calendar for this project. If it is still showing "Standard" then you have not completed step 1.3 above.



You declare project non-working time by selecting a date, or range of dates and checking the “Nonworking time” radio button.



### 3.2 Adding and setting up resources

Add the resources you expect to use using the Resource Sheet

*View => Resource Sheet*

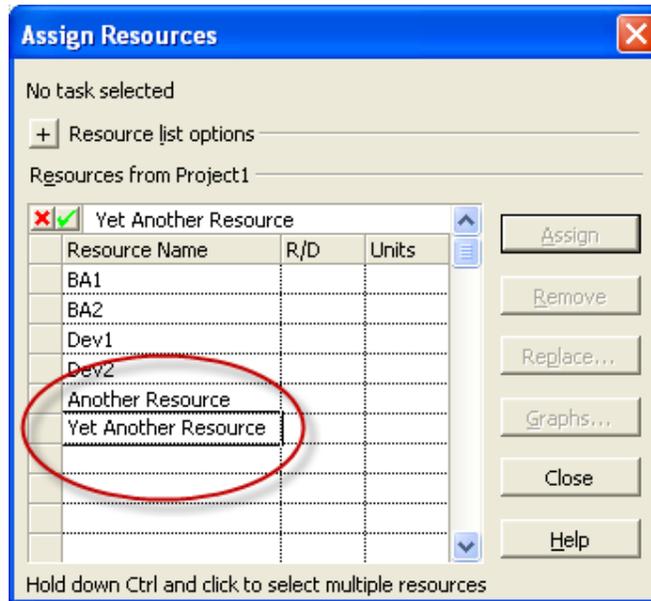
If you do not know the actual names of the resources on the project then use Dev1, Dev2, BA1, BA2 etc. However it is important to change these to real names of real people as soon as possible so that you can build in individual availability and personnel holidays.

Enter the names in the relevant place. It is also wise to enter initials at this time. The reason for this is that on a large plan where you are showing the allocated personnel next to each task it can become cluttered. When this happens you can switch to displaying Initials. Therefore it helps to have set up the initials from the beginning.

The *most important* thing to remember about the Resource Sheet entries is that you set the Base Calendar for *each resource* to be the one you set up in 1.2. This means that personnel and project holidays will be reflected in each resource's availability. It is a common mistake to forget to do this. If you forget then the problem manifests itself as MS-Project allocating work to people when they are not available, which gives false milestone and end-point dates.

	Resource Name	Type	Material Label	Initials	Group	Max. Units	Std. Rate	Ovt. Rate	Cost/Use	Accrue At	Base Calendar	Code
1	Dev1	Work		D1		100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	
2	Dev2	Work		D2		100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	
3	BA1	Work		BA1		100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	
4	BA2	Work		BA2		100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	

Do not be tempted to enter *new* resources using the Assign Resources tool:



It will let you do it, and will add the resource to the Resource Sheet, but it does not give you the chance to set the Base Calendar for each resource. You can, of course, go back into the Resource Sheet later and set the base calendar, but using the Assign Resources tool means you may forget to do so. Using the Resource Sheet to enter new resources is a good habit to develop.

### 3.3 Groups

A useful feature of MS-Project is the ability to filter displays by Resource Groups. So, for example, you can filter a Gantt chart to display only the tasks belong to a client, or an internal resource group such as Developers or BAs. In order to do this you have to set the Group up in the resource sheet.

	Resource Name	Type	Material Label	Initials	Group	Max. Units	Std. Rate
1	Dev1	Work		D1	DEV	100%	£0.0
2	Dev2	Work		D2	DEV	100%	£0.0
3	BA1	Work		BA1	BA	100%	£0.0
4	BA2	Work		BA2	BA	100%	£0.0

### 3.4 Personnel Holidays

Individual personnel holidays are entered onto the individual resource calendars. Don't try and cater for individual holidays by manipulating the Project Calendar or task start and end dates- As the project evolves and the schedule changes it will tie the project in knots.

You access an individual resource's information by double-clicking the resource entry on the Resource sheet. Note, remember to set the Base Calendar for the resource to the same as the Project Calendar before doing this otherwise the individual resource calendars will reside in the Standard calendar and will not reflect non-working time in the Project Calendar.

Resource Name	Type	Material Label	Initials	Group	Max. Units	Std. Rate	Ovt. Rate	Cost/Use	Accrue At	Base Calendar	Cod
Dev1	Work		D1	DEV	100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	
Dev2	Work		D2	DEV	100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	
BA1	Work		BA1	BA	100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	
BA2	Work		BA2	BA	100%	£0.00/hr	£0.00/hr	£0.00	Prorated	My New Project	

**Resource Information**

General Working Time Costs Notes Custom Fields

Resource name:  Initials:

Email:  Group:

Workgroup:  Code:

Type:

Booking type:  Material label:

Generic  
 Inactive

Resource Availability

Available From	Available To	Units
NA	NA	100%

Help Details... OK Cancel

Select the “Working Time” tab to access the Calendar for this user:

**Resource Information**

General Working Time Costs Notes Custom Fields

Resource Name:  Base calendar:

Set working time for selected date(s)

Legend:

- Working
- Nonworking
- Edited working hours

On this calendar:

- Edits to a day of the week
- Edits to an individual day

Select Date(s):

M	T	W	Th	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Set selected date(s) to:

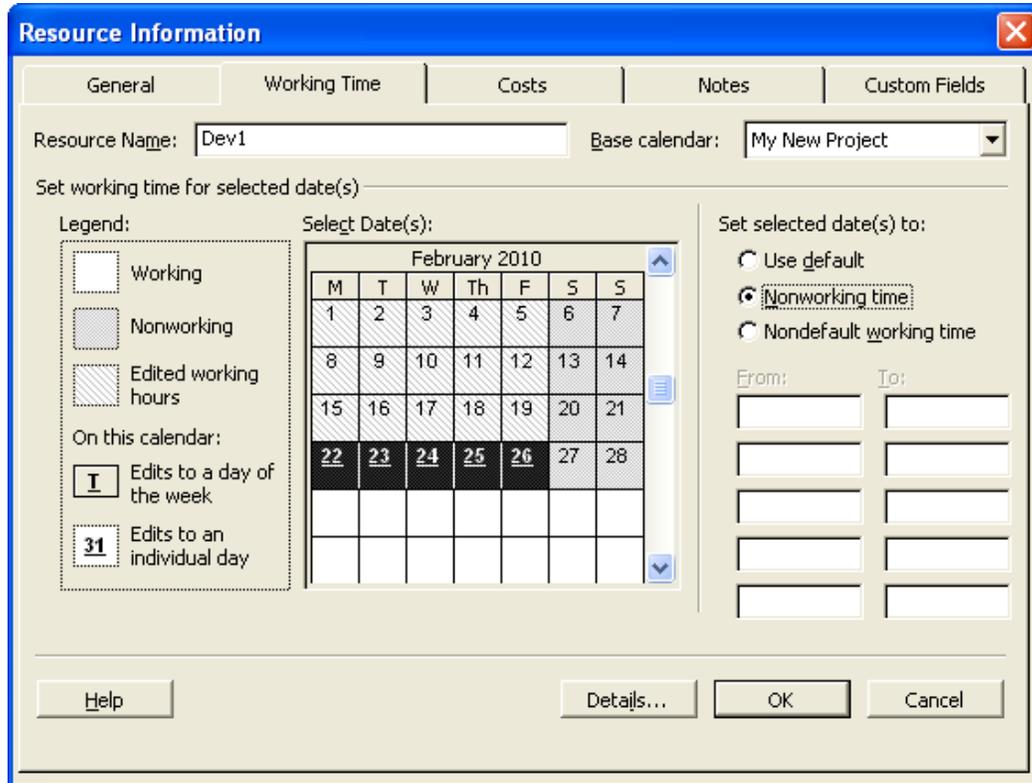
- Use default
- Nonworking time
- Nondefault working time

From:  To:

Help Details... OK Cancel

You will notice that the Calendar opens with today’s date already selected. Set non-working time for this resource by selecting dates, or date ranges and checking the “Nonworking time” radio button.

This is where you would record individuals’ holidays or other absences from the project.



If a resource is only available to the project for a finite period of time, for example they will only become available as a resource to the project on a certain date half way through the project, or can only spend one specific week on the project, *do not* attempt to use the working time calendar to show their availability. There is a much easier way- See section 3.5 below.

### 3.5 Block availability

A resource’s working time calendar assumes 100% availability and captures times within that availability that a resource *cannot* work.

But there are many instances where a resource is only available to a project with certain dates. They may start late on a project because they are still completing another assignment, or they may leave the project early to go on to some other assignment.

This declaration of actual project availability is captured on the “General” tab of the Resource details. Double-click on the resource on the Resource Sheet (*View => Resource Sheet*) to access the resource information.

Set the resource's project availability using the "Available From" and "Available To" columns in the Resource Availability field.

The dates entered can be one of three types:

Available From	Available To	Means
NA	dd/mm/yy	The resource is available from the start of the project until the date specified, which is their last working day available to the project
dd/mm/yy	NA	Resource can join the project on the date specified, which is their first working day available and is available until the end of the project
dd/mm/yy	dd/mm/yy	Resource can start on the project on the Available From date and finishes on the Available To date

You can enter any amount of entries in the "Resource Availability" table to define an individual's actual availability to the project.

## 4 Adding Tasks

The normal course of events is that a body of project tasks are added at the time the project is initially created, since you normally know the elements of the plan at that point. A good sequence of events is:

- a. Enter task blocks
- b. Enter the basic task names and durations within their blocks
- c. Enter explicit dependencies as constraints
- d. Assign Resources
- e. Level the plan
- f. Arrange tasks using Priority
- g. Enter milestones

Like anything within MS-Project, it is possible to do these things in different orders and at different times. Sticking roughly to the order above is likely to yield a project plan that is easier to maintain as changes naturally occur through the evolution of a project.

### 4.1 Enter task blocks

Before entering individual tasks it is worth spending a small amount of time thinking about the high-level task blocks. For example, in a standard waterfall development there are likely to be several stages:

- Analysis
- Development
- System Testing
- User Acceptance Testing

However you might be planning a large project, with several phases and several separate mini-projects within each phase. Try to reflect the broad project hierarchy of work in this manner using blocks and don't be afraid of nesting them several layers deep in order to get the arrangement right.

	Task Name	Duration	01 Feb '10					08 Feb '10					15 Feb '10										
			M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	My New Project	1 day?																					
2																							
3	Phase 1	1 day?																					
4																							
5	Analysis	1 day?																					
6																							
7	Development	1 day?																					
8																							
9	System Testing	1 day?																					
10																							
11	UAT	1 day?																					



Task Name	Duration	01 Feb '10					08 Feb '10					15 Feb '10									
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T		
My New Project	1 day?																				
Phase 1	1 day?																				
Analysis	1 day?																				
User Workshop 1	1 day																				
User Workshop 2	2 hrs																				
Development	1 day?																				
System Testing	1 day?																				
UAT	1 day?																				

Don't forget to indent tasks that belong to a task block. By default, MS-Project will auto-indent all tasks to the same level as the task above, which means that when entering tasks beneath the task block name they will all be indented to the same level. When you have finished entering the tasks for a block, select them all and click on the indent button.



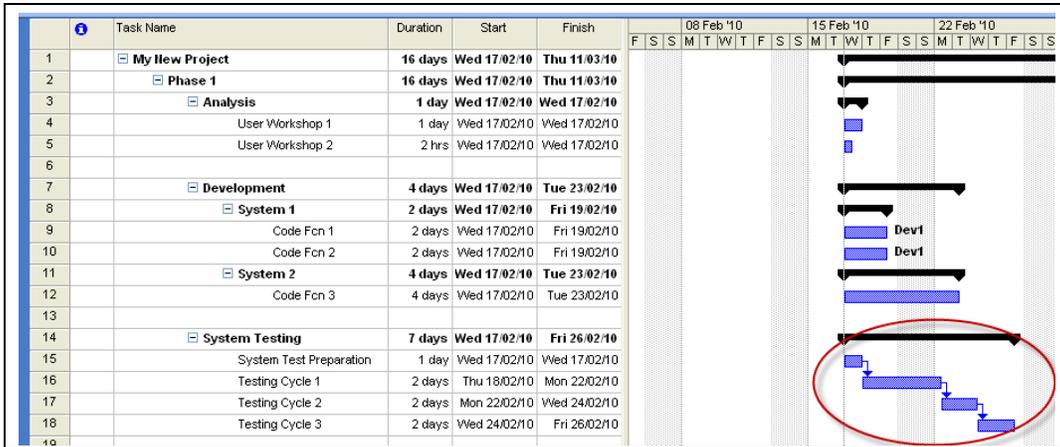
Task Name	Duration	Feb '10					08 Feb '10					15 Feb '10									
		T	W	T	F	S	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
My New Project	1 day?																				
Phase 1	1 day?																				
Analysis	1 day																				
User Workshop 1	1 day																				
User Workshop 2	2 hrs																				
Development	1 day?																				
System Testing	1 day?																				
UAT	1 day?																				

Try to complete entering the tasks and durations before attempting to add constraints or milestones.

### 4.3 Enter explicit dependencies as constraints

By far the biggest problem encountered when entering tasks is overuse of constraints (i.e. Predecessors). You aim should be to reduce the amount of constraints in the project plan, to the absolute minimum.

A constraint should only be used when there is an actual reason why one piece of work can only be started after a preceding piece of work. An example would be the System Testing cycles- Clearly you cannot start a second testing cycle until the first has been completed, so you would expect to make Cycle 1 a predecessor of Cycle 2.



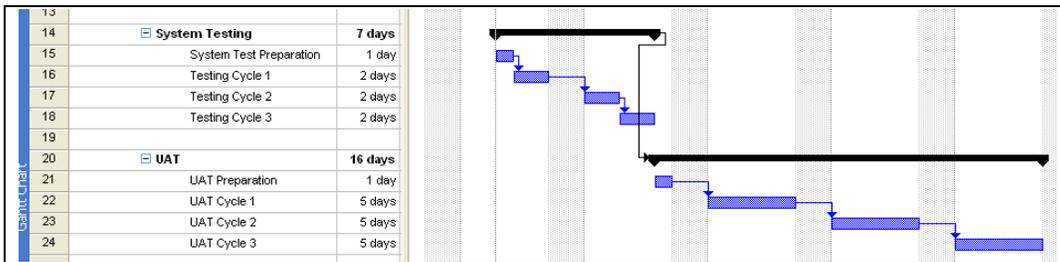
However if you have two separate items of development that are not linked technically and you wanted to ensure that one was done before the other, you should NOT use a predecessor to make that happen in the plan.

*This is the single biggest problem that users have with MS-Project for anything but the most trivial plans.*

Use of constraints to order tasks ties the plan in knots and reduces the opportunity for efficient allocation of resources. Additionally, and perhaps more importantly, in the real world people rarely completely finish one task before starting another- This means that a plan based on a constrained sequence of events will elongate artificially to try and ensure that things are done in a set order. Let tasks “float” in the plan. This may be a bit of a leap of faith at the moment- I will return to this in section 6.3 Task Start Dates .

**Remember!**  
Only link a task to a predecessor when there is a real and explicit reason to do so!

Where there is a real dependency among a group of tasks, for example we can only commence UAT when System Testing has been signed off. It is far better to put the dependency between the task groups than between individual tasks.



This means that the broad dependency of one activity on another is retained, but the individual tasks can change or be moved around without having to update loads of constraints.

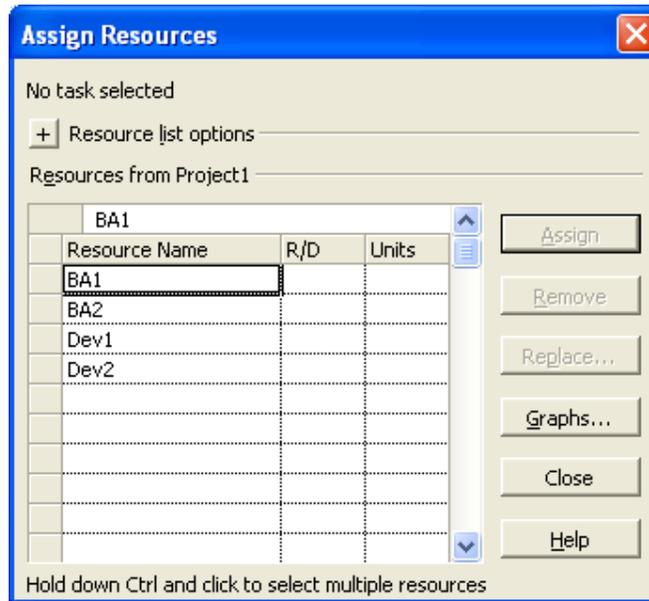
## 4.4 Assign Resources

Once you have entered all the tasks into the plan, it is time to assign actual resources to the tasks. Do not worry at this stage about the scheduling and ordering of tasks.

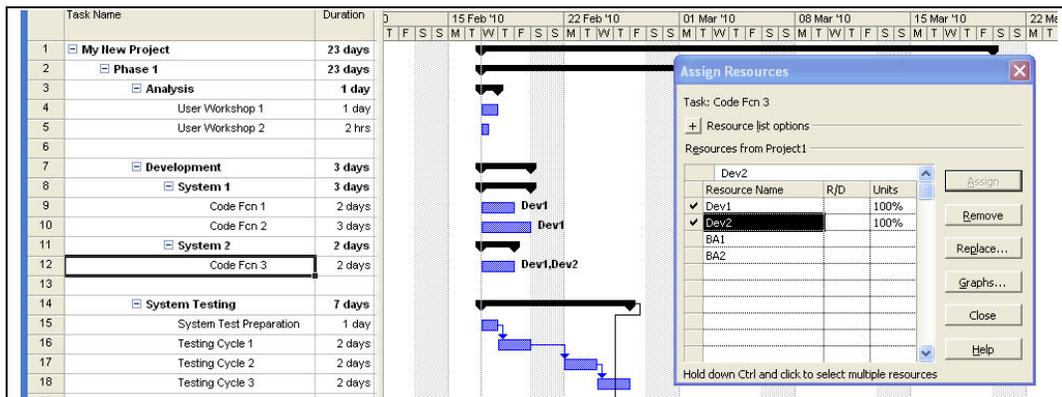


It is easiest to use the “Assign Resources” panel to assign since it can be left on screen as you move between tasks. The easiest way to bring up the Assign Resources panel is to use the Resources button on the toolbar.

Select a task on the plan and click on the “Assign Resources” button.

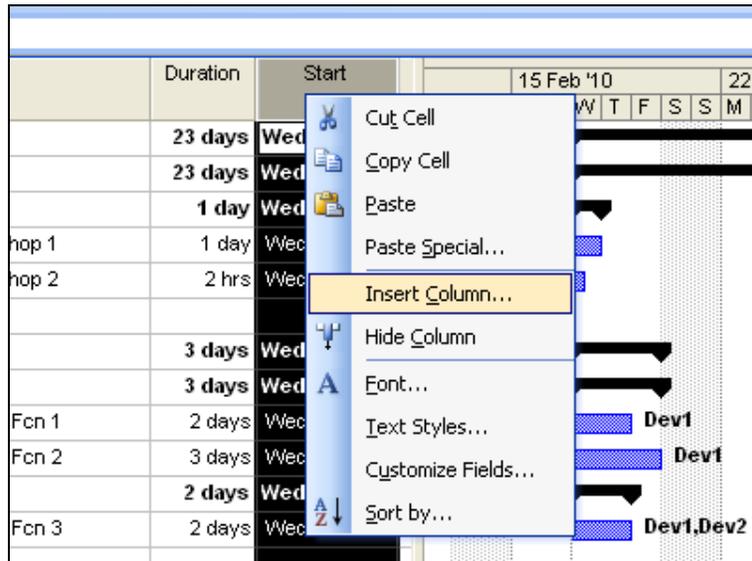


You can then select an individual or group of resources on the “Assign Resources” panel and assign them to the task by pressing the “Assign” button.

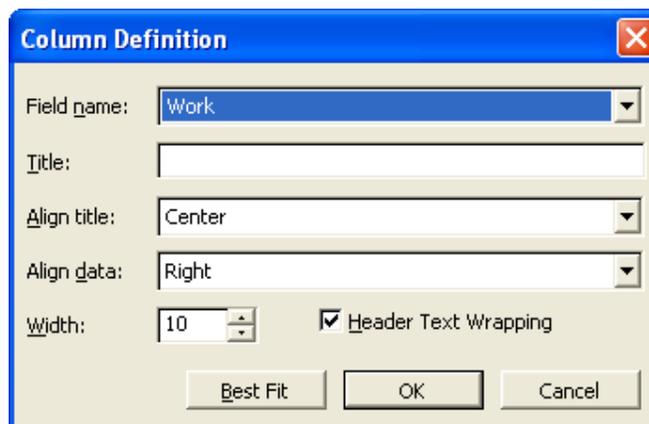


Remember that once you start adding resources to a task, its duration will change but the actual effort remains the same. If one person can do a task in four days, two people will be able to complete it in two days, but it still requires four man-days of effort.

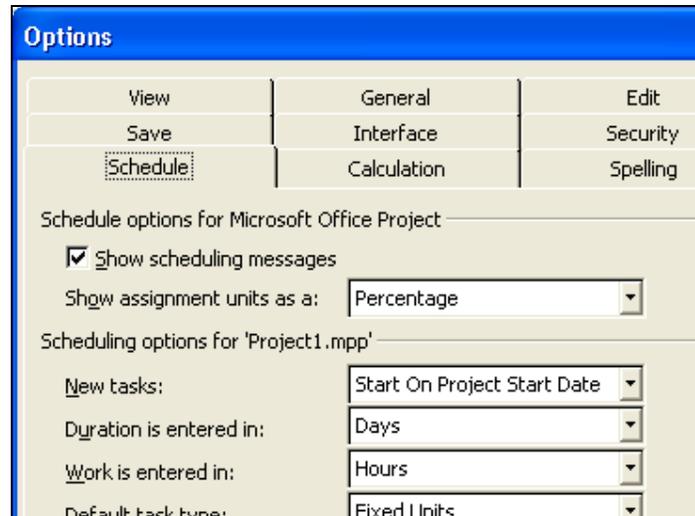
If you need an easy way to see the actual underlying effort for tasks in addition to the duration, then add the 'Work' column to the left-hand panel. Right-click on the "Start" column title and select 'Insert Column...' from the dropdown menu.



Select the Field name 'Work' in the Column Definition panel and press OK.



You will notice that entries in the 'Work' column are measured in hours by default. If you want to change this use Tools => Options and select the 'Schedule' tab.



Alter the 'Work is entered in:' field to reflect the required work units.

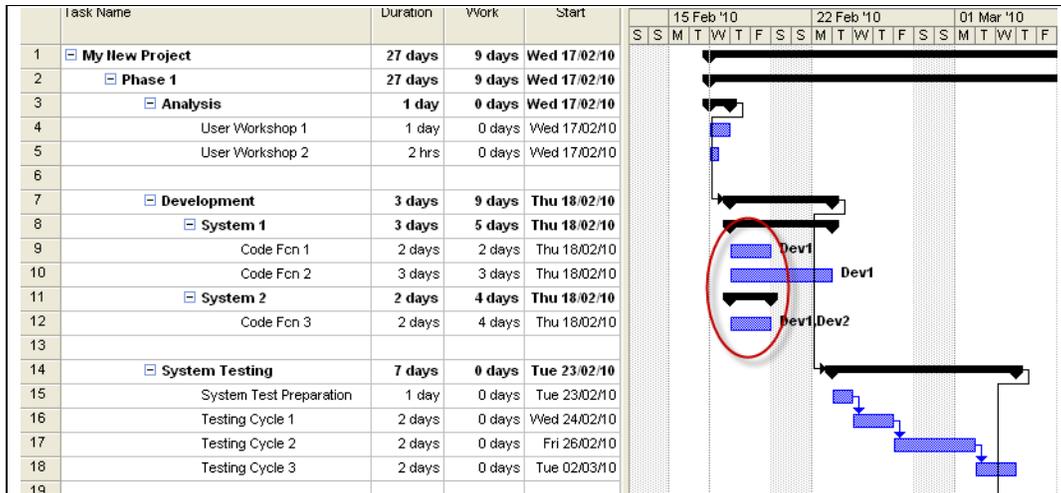
Code Fcn 2	3 days	3 days
<input type="checkbox"/> System 2	2 days	4 days
Code Fcn 3	2 days	4 days
<input type="checkbox"/> System Testing	7 days	0 days

The table shows task durations in days. A red circle highlights the '2 days' value for 'Code Fcn 3'. To the right, a Gantt chart shows task bars. 'Code Fcn 3' has a shorter bar than 'Code Fcn 2' and is assigned resources 'Dev1' and 'Dev1,Dev2', illustrating how resource assignment affects task duration.

Notice that in the example above, even though 'Code Fcn 3' was entered with a four-day duration, it has automatically become a two-day duration through the addition of two resources. However the 'Work' column has remained at four days.

## 4.5 Level the Plan

Resource levelling is the action of going through the plan and ensuring that resources are not over-allocated (since someone cannot be doing two things at once). Usually this means that all the resources are checked to ensure they are only scheduled to work one day (i.e. 7.5 hours) per actual day. Initially, following task entry and resource allocation, resources will not be levelled, tasks are allowed to be piled up on resources with no regard to how many hours are being allocated to resources even within one day.

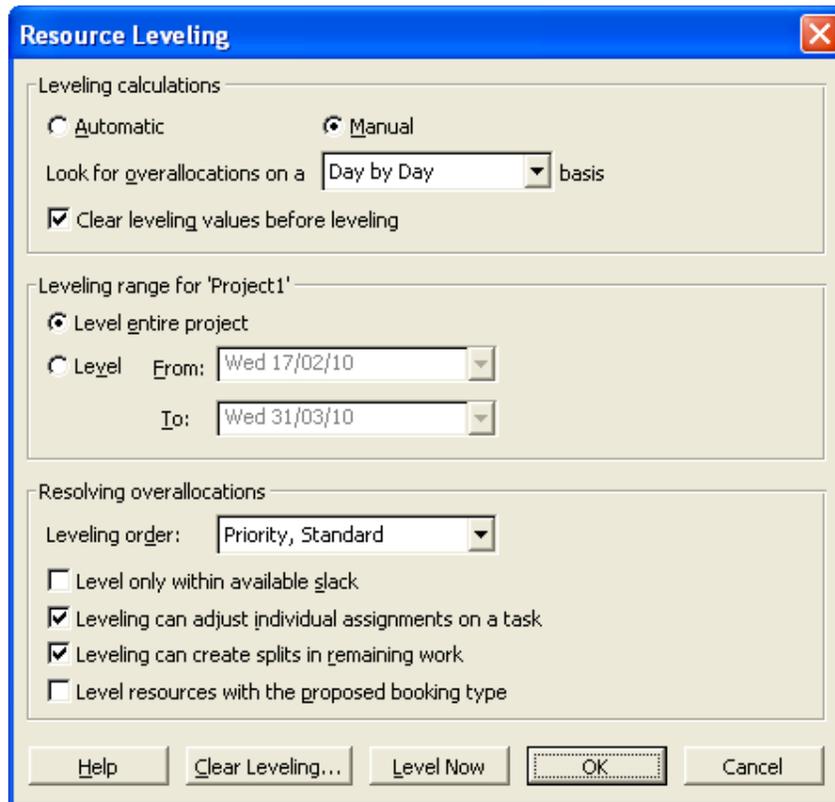


In the example above on Thursday and Friday Dev1 is supposedly doing a day's work on three separate tasks at the same time, equalling 22.5 hours work *per day*! Clearly this will not reflect what happens in the real world. However the more important effect is that since MS-Project is assuming that both tasks can run in parallel, the overall duration of the System 1 task block remains at three days (the duration of the longest tasks within it).

**Remember!**  
An unlevelled project plan will probably have a false end-date

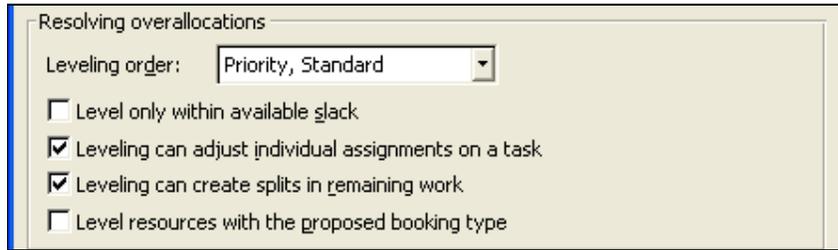
By levelling the plan, MS-Project will sort out all over-allocations. Levelling is an extremely powerful tool and should be used liberally. Get into the habit of levelling the plan after any change to tasks or resources.

To level the plan, use *Tools => Level Resources*



Note that it is possible to set Levelling to happen on an automatic basis. I recommend that you do not do this until you are completely happy with the effects levelling has on your plan. I find it easiest to make a block of changes to a plan and then level once at the end. Automatic levelling, if enabled, will be applied after every single change.

Ensure that the items are set per below:



Leave all other settings at default.

Press the 'Level Now' button.



Now, in the example provided, MS-Project has moved and split tasks to ensure that the Dev1 resource is only doing one day of effort per calendar day.

Given the tasks and the resource available, MS-Project has scheduled the tasks to produce the shortest possible duration. Dev1 has been given a total of seven days work on this plan and therefore the shortest overall duration of the work requiring his capacity is seven days, whereas before levelling the overall duration of the Development tasks was three days (not including the weekend).

It is interesting to note that it has also put a split into the Code Fcn 3 task. It has allowed Dev2 to do as much as allowed on the task (two days) and then the task has to wait for Dev1 to become free after their first task. At that point, Code Fcn 3 is completed and Dev1 is allowed to move back to their second task.

**Remember!**  
It is only through very frequent use of Resource Levelling that you will have an accurate end-date!

## 4.6 Arrange Tasks Using Priority

MS-Project tries to arrange tasks in a plan according to the best fit and availability of resources assigned. If you do not constrain Project through overuse of predecessors, it will schedule tasks to be started according to an internal algorithm (largest tasks first, smallest tasks last). However, in the real world you will often want to undertake blocks of tasks in a certain order.

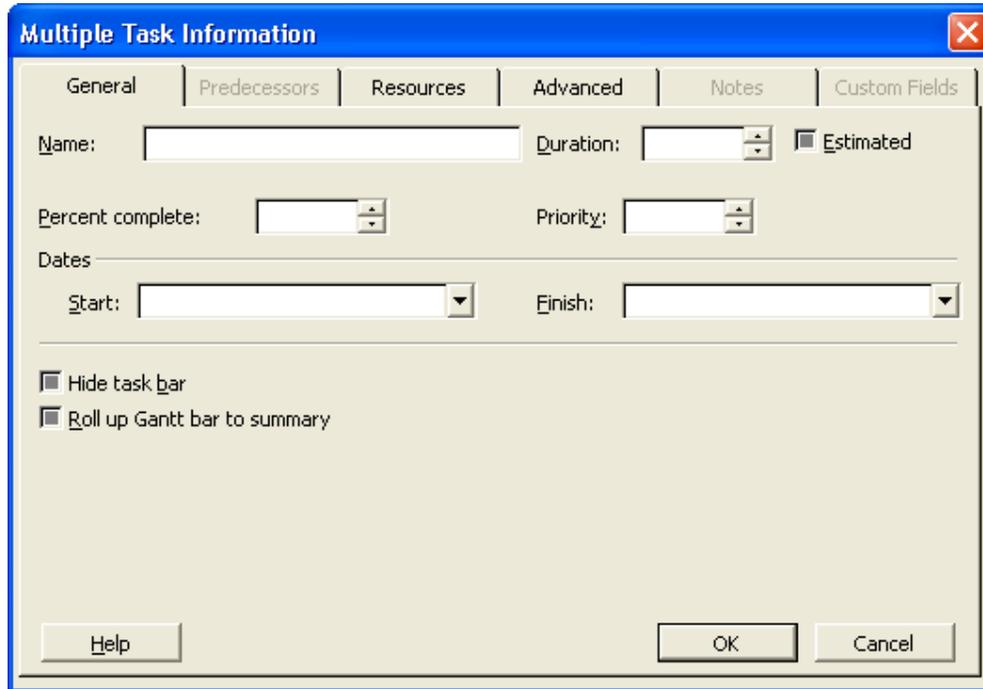
You set the scheduling order through the use of the Priority field. In MS-Project, 999 is the highest possible priority and 1 is the lowest. It is tempting to try and use Task Priority to ensure all tasks within a block are done in a strict order. Try to resist this temptation. As documented in section 4.3 (Enter explicit dependencies as constraints) tying a project up with constraints that are not real dependencies is bad for the project and using task priorities to achieve the same result is nearly as bad. It is far better to let tasks “float” and allow MS-Project to schedule them where it sees fit until actual work is done, which then locks down a task start date. I will return to this in section 6.3 Task Start Dates.

In spite of this a good use of Task Priority is to ensure certain *blocks* of work are started before others by setting all the tasks in one block to a higher priority than other blocks. In the example below Ms-Project has been allowed to schedule tasks according to its own best-fit. However suppose it is beneficial for some reason to ensure that all the work for System 2 is completed first...



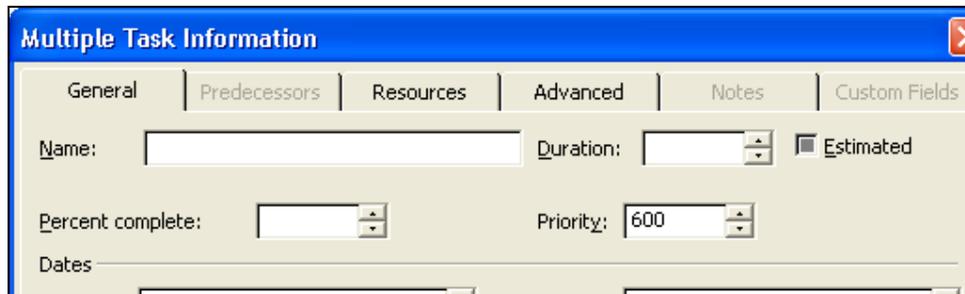
To achieve this we need to ensure we set the priority of the tasks in the System 2 block higher than the other items which ensures the task 'Code Fcn 3' is completed as soon as possible rather than being completed when Dev1 is free from their first task.

Unfortunately, MS-Project does not allow you to set a priority at a task block level- you DO have to go into each item in the block and set the priority. Fortunately, MS-Project allows you to multi-select a group of tasks and amend them in one operation. Select the tasks, right-click on the selected group and bring up the 'Task Information' panel.



By setting the priority of the tasks you affect the way they are scheduled (and levelled) by MS-Project. In the current example, setting the priority of all of the 'System 2' tasks to 600 instructs MS-Project to attempt to schedule the completion of all tasks in System 2 *before* other lower-priority tasks (all tasks have a default priority of 500 when you enter them in MS-Project).

Enter the priority and select ok.



To see the effects of the priority changes ensure you level the plan immediately (see section 4.5).



Instead of splitting 'Code Fcn 3' in order to wait for Dev1's availability, Dev1 is now allocated to it from the beginning and only when 'System 2' is complete is Dev1 allowed to begin work on other tasks. The overall duration has not changed, but the order of task start and completion has altered.

Use task priorities to ensure tasks are started/completed ahead of other tasks. Use this technique instead of setting task dependencies between two tasks that are not actually dependant.

## 4.7 Enter Milestones

Use of milestones outside of task blocks means it becomes easy to tell at-a-glance when key events are happening in the project. These should also be reflected in project progress reports therefore their transparency in the plan will assist creation of that report.

A milestone is just like a normal task, but they have zero-duration. Enter a milestone name in the same way as you would enter a task name, and put in '0d' as the duration. It is not necessary to assign resources to milestones, however it is entirely possible to do so if a certain resource is responsible for a milestone (especially for external constraints, see section 5.1 External Constraints and Milestones)

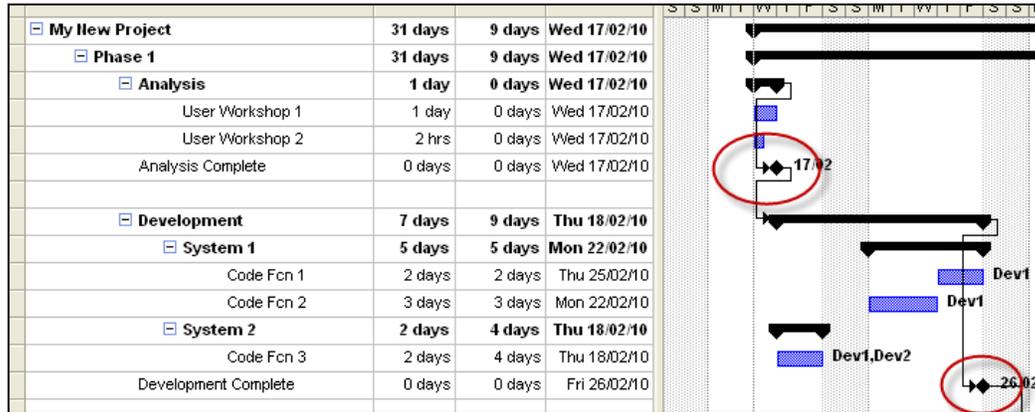
Once a plan has a liberal number of milestones, and progress towards each milestone is measured and reported, it becomes a lot easier to detect early on in the project whether the schedule is slipping. Additionally, early effort to bring a project back on track to hit the next milestone automatically brings all of the milestones back on track, including the delivery date. This means that outstanding work and effort is not left to pile up at the end of the project. This in turn means that there is not a mad panic at the end to try and fit too much work into too little remaining time, which inevitably results in a slippage at the end.

Try to add a milestone to the end of relevant task blocks, rather than against individual tasks. For example you might want to add the following milestones:

- Analysis Complete
- Development Complete
- System Testing Complete
- UAT Commence
- UAT Complete
- Go-Live

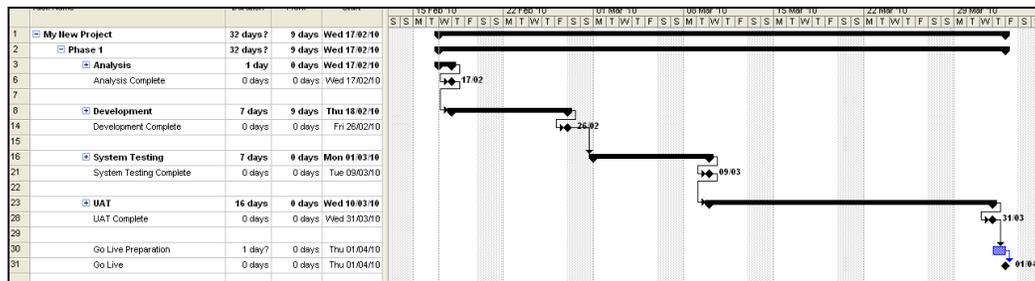
These are all milestones that relate to an entire block. Just make the block a predecessor of the milestone to link them together.

Once you have milestones between blocks it is useful to change the constraints so that subsequent work depends on preceding milestones instead of preceding blocks.



This makes the plan easier to maintain because it is easier to insert tasks or task blocks before a milestone- All you need to change is one constraint to make the milestone dependant on inserted blocks and everything else continues to “work”.

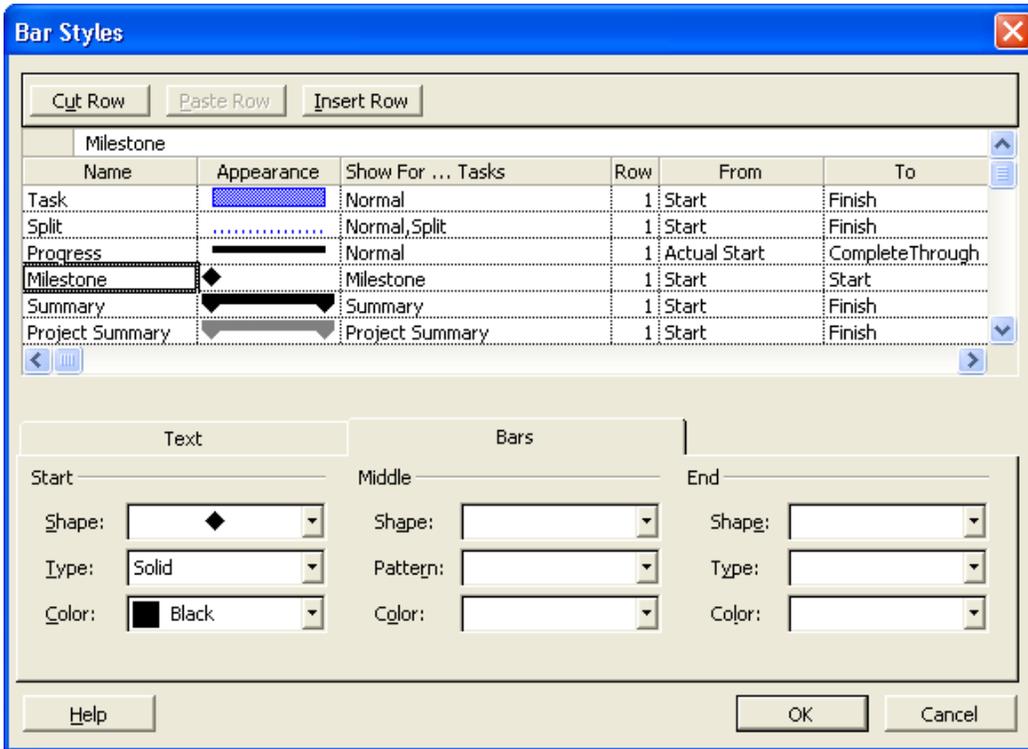
The other beauty of this arrangement comes when you “roll up” all the task blocks (compress all the tasks on the plan so that it only shows the higher-level task block item by clicking on the small ‘minus’ symbol next to the block)



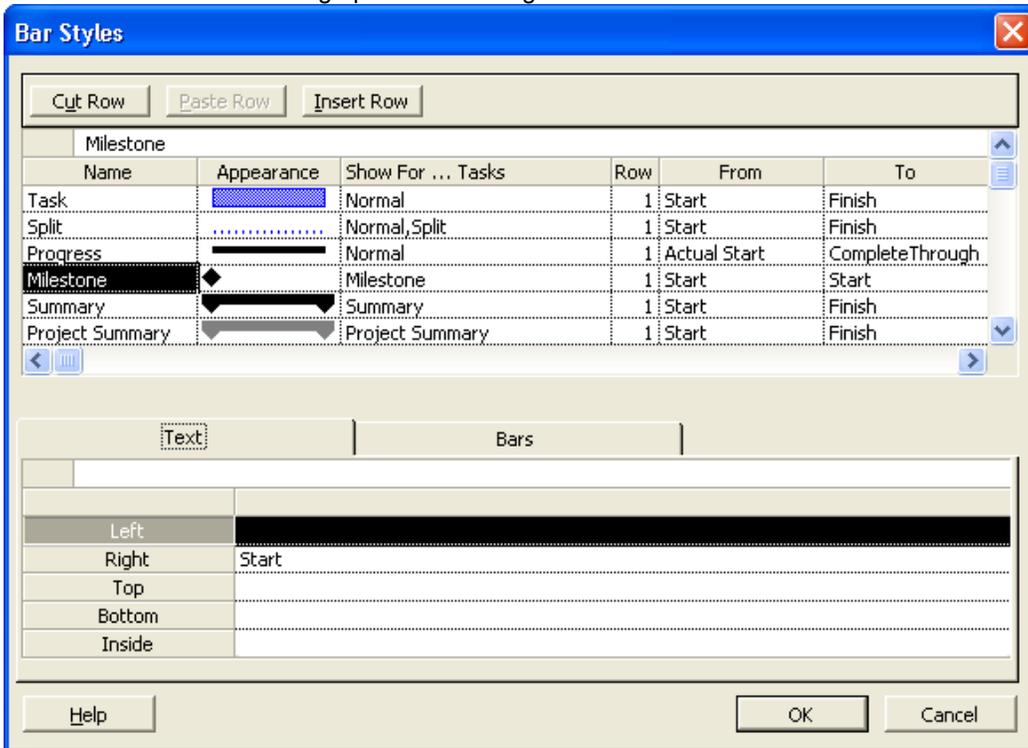
Now the plan shows key task dates only, plus the milestones that are external to the task blocks. It hides the complexity of the underlying tasks which is very useful when reporting to clients.

One nice effect is to include the milestone name and date against the actual milestone on the plan. This is a “Bar Format” option.

Using *Format => Bar Styles*, bring up the Bar Styles panel and select the 'Milestone' item.

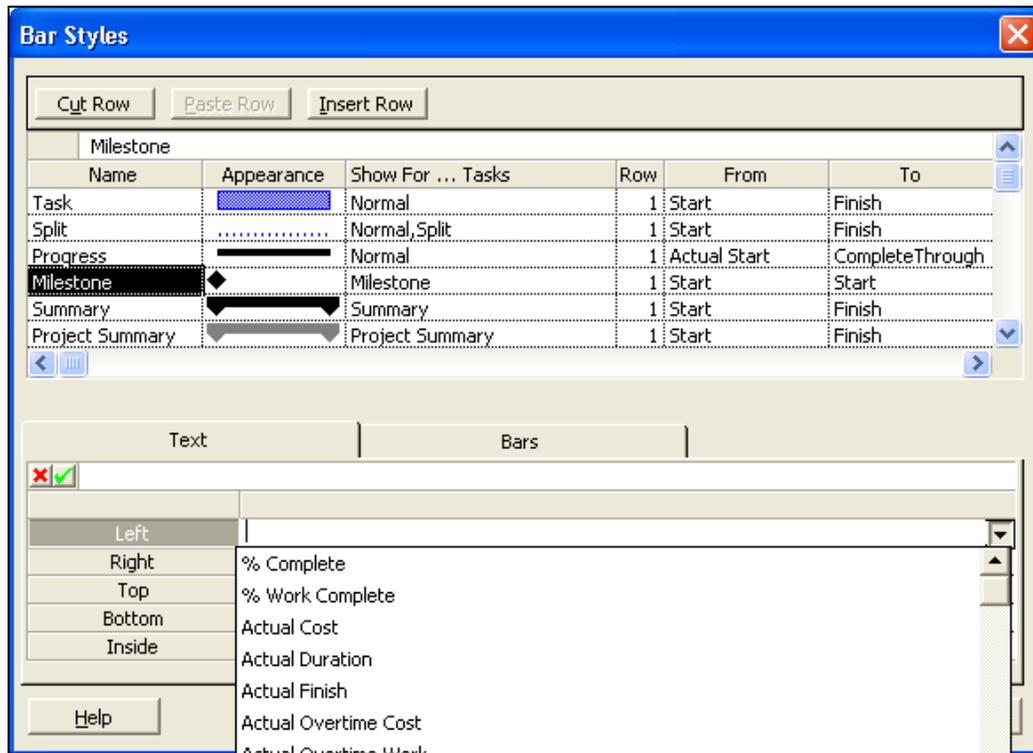


Click on the 'Text' tab to bring up the text settings for the milestone item.



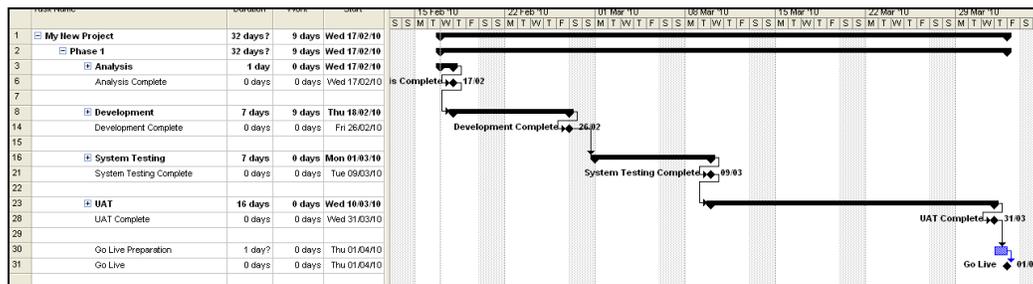
This panel lets you automatically place text around the milestone symbol on the plan. By default milestones automatically place the Start date to the right of the symbol.

Select the 'Left' item and open the dropdown



Select 'Name' from the dropdown list and click 'Okay'

This adds the tasks name to the left-hand side of all milestones on the plan. This considerably eases the readability of the plan, whether rolled up or not.

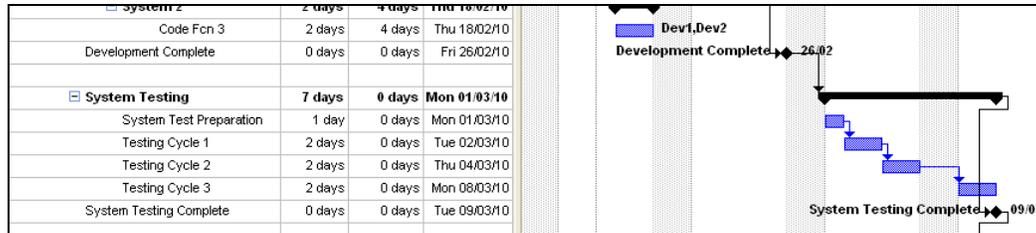


You can use the same technique on the 'Task' bar-style to make the plan more readable- You may possibly want to change the default action of MS-Project of displaying all allocated Resource names next to the tasks, to perhaps displaying only their Initials.

## 5 Other Scheduling and Planning

### 5.1 External Constraints and Milestones

Sometimes scheduling and planning the ordering of tasks using the priorities is not enough and it becomes necessary to fix a start date for a task for reasons outside the plan. For example, in our example project System Testing commences as soon as the development is complete.



It might be the case that the System Test environment is not available until the following week and thus the System Testing tasks cannot commence until then. No amount of modification of the project tasks will force the system testing to commence on Monday 8<sup>th</sup> March. However this is one of the occasions where there *is* a real constraint- the testing cannot start until the environment is available.

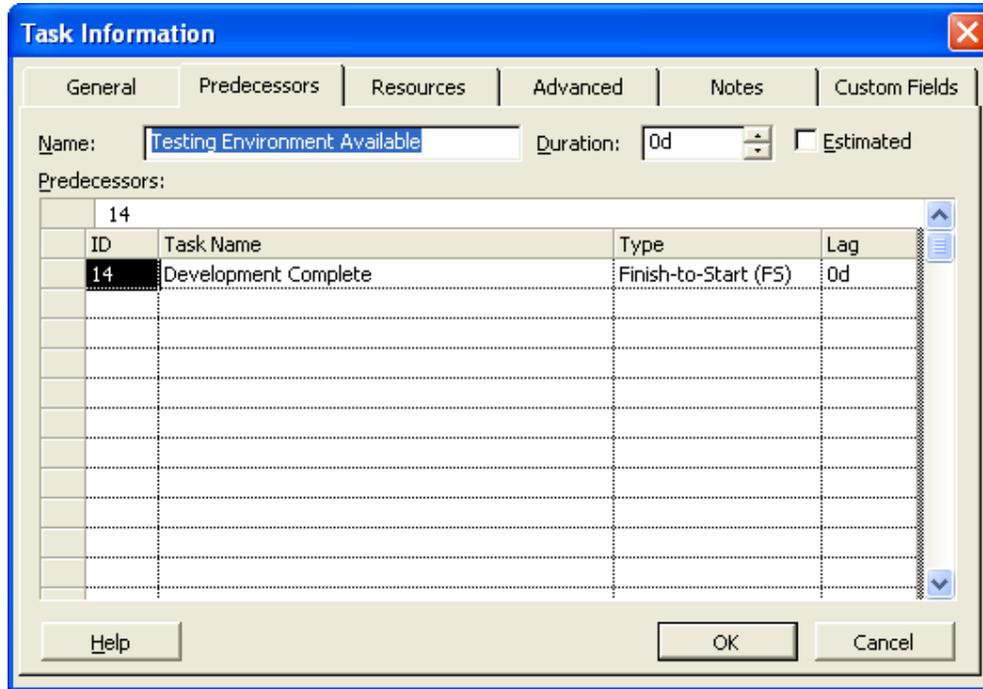
There are several ways this could be shown in the project, including allowing a one-week slip between the end of Development and the start of System Testing, or setting a Must Start On constraint on the System Testing block or first task. This kind of problem is an example of an external constraint- this project cannot make use of an external resource (the testing environment) until it is available. External constraints are common; you may not be able to commence some analysis until a third party has sent some documentation, or you may not be able to process a file until it has been received from another system. The list is endless.

However the best way of showing an external constraint is with a new milestone. In this case we can set up a milestone to signify when the testing environment will become available.



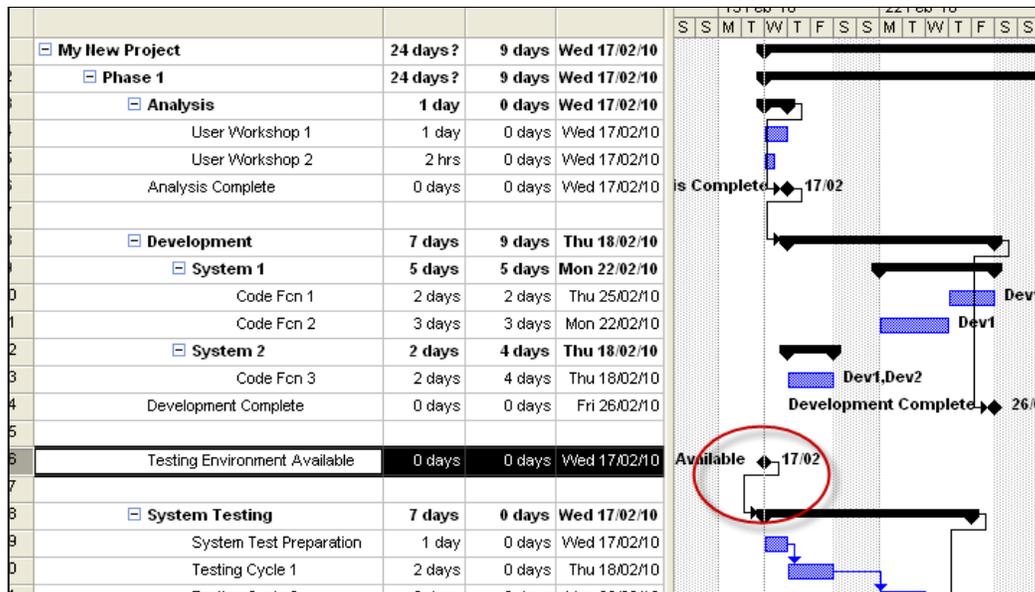
MS-Project tries to be helpful. Because we inserted the milestone task between two linked tasks (the Development Complete milestone and the System Testing task block) it has assumed it is an intermediate step. However that is only partially true- The start of testing *does* depend on the Environment Available milestone, but that milestone *does not* depend on the end of development, it is entirely external and its actual date will depend on circumstances outside the control of the project. Therefore we must break the link between the end of development and the Testing Environment Available milestone.

Double-click on the Testing Environment Available milestone and select the Predecessors tab.



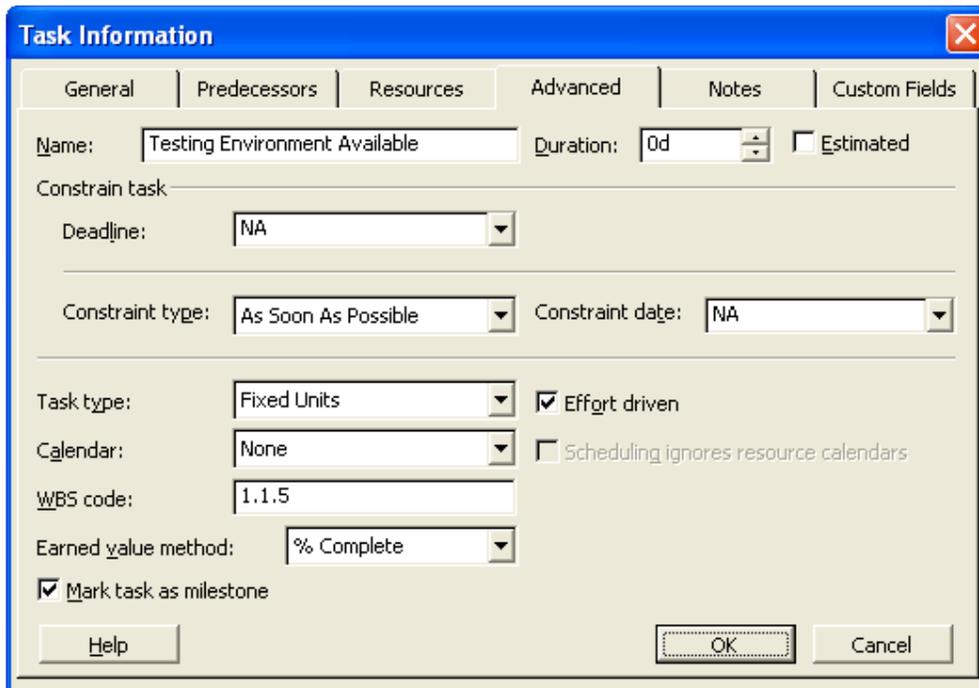
Position the cursor in the ID column of the task you want to remove and press the delete key on the keyboard and the dependency will be removed. Press OK to return to the project plan.

The dependency has been removed, but this had had the effect of putting the Testing Environment Available milestone at the start of the project together with the rest of the testing tasks which now start before development has completed!

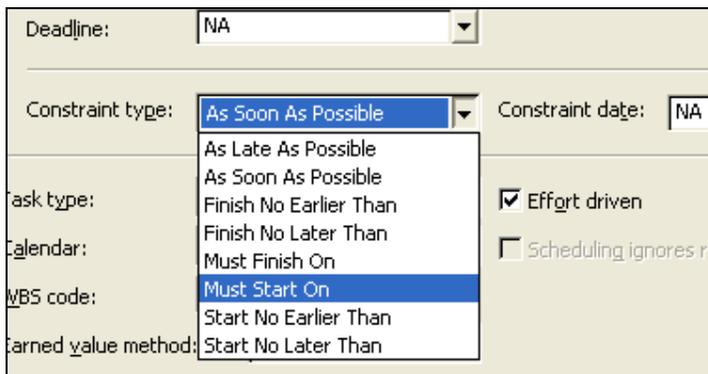


This is because MS-Project has no information about when this milestone might occur. In this example we have been told that the testing environment will be available on Monday 8<sup>th</sup> March. So we can now apply a Must Start On constraint to the milestone task.

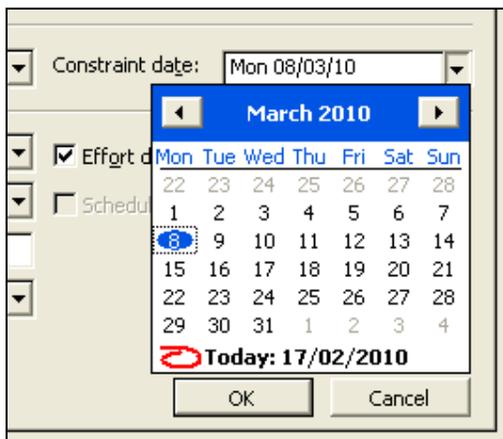
Double-click on the milestone task and choose the Advanced tab.



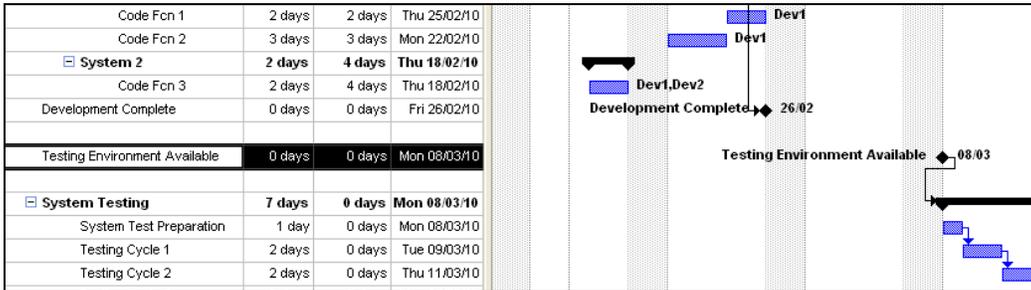
In the Constraint Type field, select “Must Start On”



In the Constraint Date enter (or select) the date to be applied.

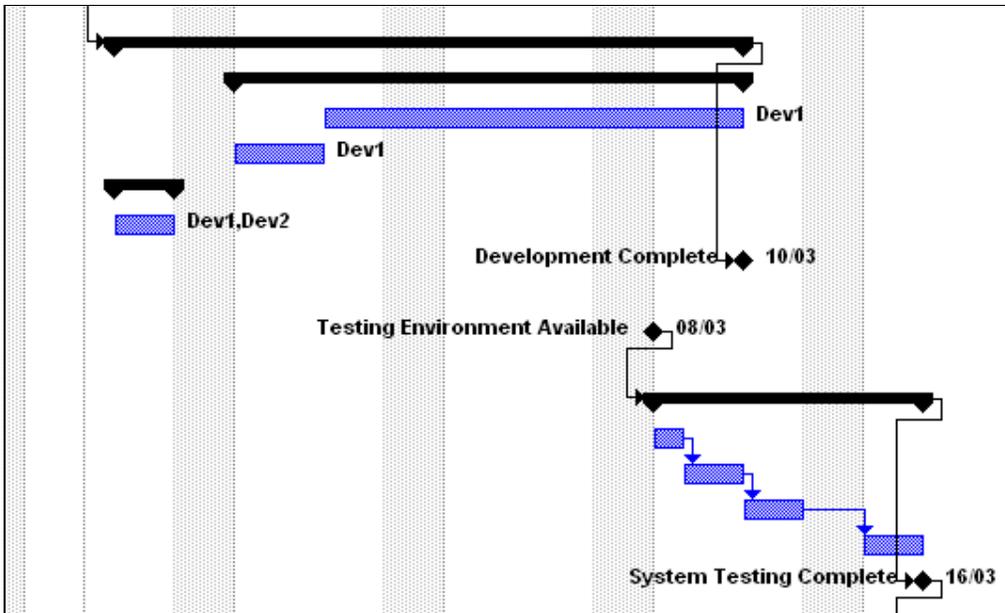


Press OK to return to the project plan. The Testing Environment Available milestone has now moved to its correct date and because it is a predecessor of the system testing, that has moved to occur after the milestone.

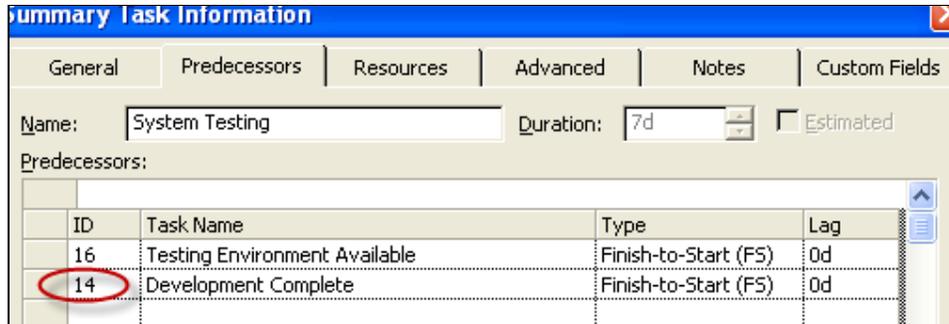


Unfortunately, we have now broken an internal constraint; even though the System Testing tasks cannot start until the environment is available it also cannot commence until the development has completed. On the current plan it does not look as if that is a problem since the development naturally completes well in advance of the new system testing date.

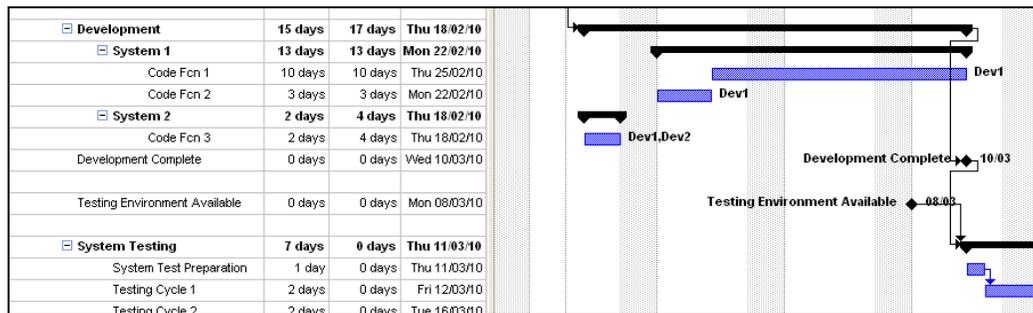
But let us suppose that after the plan is drafted the work to complete Code Fcn1 is re-estimated to be 10 days instead of two. This has the effect of delaying the completion of development.



Unfortunately, since the system testing was only dependant on the environment becoming available it looks as if the system testing can commence before the development can complete. We need to link the start of testing to the completion of development as we did before the introduction of the external constraint. Using the Predecessors tab of the System Testing task block, add the Development Complete milestone as a predecessor of System Testing



The additional constraint has now been added to the project plan and the testing tasks are moved accordingly.



This kind of handling of external dependencies and milestones helps us track the effect of delays in those milestones and constraints. According to the plan the system will go-live on 15<sup>th</sup> April and that is dependant upon the system test environment becoming available as planned on 8<sup>th</sup> March (albeit with a few days slack, useful to remember) and the development completing on 10<sup>th</sup> March. If we subsequently hear that the testing environment will not now be available until 15<sup>th</sup> March we can easily modify the date of the milestone to the new date and immediately observe the effect it has on the downstream milestones and go-live slips from 15<sup>th</sup> April to 19<sup>th</sup> April. This is information the Project Manager needs in order to manage external constraints and third parties.

## 5.2 Baselining

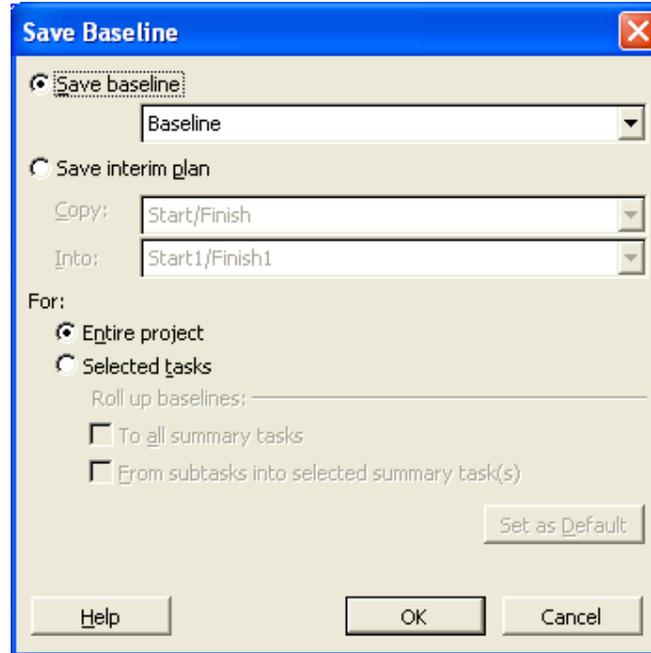
When you have completed setting up the project and you are reasonably certain that it contains all tasks and all internal and external dependencies, in other words the plan is complete and you are ready to start doing the actual work and tracking against the plan, you should first *baseline* it.

A baseline is like a snapshot of the project including all the tasks and all their start and end dates. It is stored internally within the project. When work gets underway and you are tracking your actuals (i.e. you are recording the actual start and end dates of tasks and how far through tasks the project team has gone) things inevitably start to move around on the plan as tasks get delayed or completed early. Without referencing the original plan it is sometimes difficult to see the effects of cumulative movements at a glance, so MS-Project provides a different view of the project, called the Tracking Gantt that includes the baselined information

and the current information. It becomes easy to see how tasks are moving against the original plan and additionally the current Critical Path of the project (more on this later).

You can baseline and re-baseline your project at any time using the Save Baseline screen.

*Tools => Tracking => Save Baseline*



Though there are many options, the simplest is to leave the settings at default (as shown above) and press the OK button. The baseline will be saved.

## 6 Tracking Against a Plan

A project plan has two uses:

- Its first use is as a plan of how things can be done in the future to deliver the objectives (e.g. put a new system live). By entering all the tasks, all their estimated durations and applying all the resource one expects to apply, the plan will deliver to the Project Manager the start and end dates of all the tasks and milestones in the plan, including the final delivery milestone (assuming you have put one in the plan!)
- Once the plan is underway and the tasks are being worked on, the plan becomes the way the Project Manager determines whether the project is still on track to deliver against the original planned dates. This is called tracking against the plan and is achieved by constantly updating the plan with actuals (i.e. things that really happen, start and end dates and time spent on the actual work) and letting MS-Project recalculate the plan and replan the dates of the remaining tasks

The first three sections below discuss the main tracking options available to the Project Manager before moving on to describing the sequence of events a Project Manager will exercise frequently in order to track against a plan in section 6.4 Putting Tracking into Practise. The latter section brings all the techniques together and offers a set method of working through the complex task of tracking a project's current position against the project plan.

### 6.1 Tracking by Percentage Complete

There are a number of ways of telling MS-Project what is actually happening, the commonest of which is by declaring the "Percentage Complete" for tasks as they occur. In other words, the Project Manager looks at tasks as they are being completed by the team and periodically updates the project by declaring tasks to be a certain percentage complete. When a task is set as 100% complete it is deemed to be *actually* complete and work may begin on other tasks.

This method is fine for simple projects where the tasks are linearly constrained, i.e. all the tasks locked into predecessors in a set order, and that is why it is such a common mistake to apply too many constraints in a project plan to force the tasks into the "right" order.

Unfortunately there are two very serious issues with this approach:

- For all but the most trivial of tasks, especially in the field of software development, tasks do not always get to 100% complete before the development team starts on the next task. If the plan is locked into a strict order of tasks, one following the other, the plan quickly becomes unworkable as a tracking device. Typically Project Managers then try to rebalance a plan by moving tasks and adding more constraints to continue to force the plan to reflect reality. Often the plan will eventually become so difficult to maintain that it is thrown away, which means the end of the Project Manager's ability to determine if their project is still on track
- It is notoriously difficult to get an accurate answer to the question "What percentage of this task have you completed", especially from developers. More importantly, this completely hides any effects resulting from the original estimates being incorrect. If a developer is working on a four day task and they reply "50%" when you ask them how far through they are, it could be because they really are half way through, or it could be that they have spent two days on a four day task and therefore they must be 50% of the way through, which is easier to report than "I have spent two days on this four day job, but really I have only just started, it is now looking more like a ten day job"

Sooner or later you will have to get acquainted with the (only slightly) more complicated, but much more accurate method of tracking actuals, Tracking by Work.

## 6.2 Tracking by Work

All tasks have a number of attributes that you need to know in order to track their progress. Knowing what was originally estimated for the work in terms of effort and or duration is interesting but, as we saw in the Tracking by Percentage Complete section above, is ultimately useless when trying to figure out where things stand *now*.

What you really need to know is:

1. How much time has been spent on the task to-date
2. How much more time needs to be spent on it to complete it

Of course, from this information it is easy to determine a percentage complete, however by asking these two questions we can quickly see whether the task is taking longer than it should. This information is what you *should* be asking for from the project team at all times. In an ideal world you can get the answer to the first question from a timesheet or other time tracking system since it is unlikely that an individual will know, for anything other than the smallest tasks, how much total time they have spent on it.

However, whether you obtain it from a timesheet system or by keeping your own tally, it is the Project Manager's responsibility to know how much total time has been spent on a task. Fortunately MS-Project provides an easy way for you to record this information.

MS-Project maintains three numbers which are useful for this (amongst many others):

Value	Meaning
Work	The expected <i>current</i> total effort to complete the task
Actual Work	The actual amount of effort that has been expended on the task to-date
Remaining Work	The <i>currently</i> expected effort remaining to complete the task

These can (and should) be added to the project columns- In fact we have already added "Work" in section 4.4

Assign Resources when looking at the difference between Work and Duration. You should add these to the plan right at the start (though they can be added at any time thereafter) and enter the answers to the key questions above into the relevant columns in the plan (rather than by maintaining the Percentage Complete field within each task).

### 6.3 Task Start Dates

We have already looked at the difficulties of maintaining a plan where the individual tasks are locked in a specific order nose to tail and in section 4.3 (Enter explicit dependencies as constraints) it was recommended that tasks are left to “float”.

Allowing tasks to float allows MS-Project the maximum flexibility when replanning to cope with resourcing and other scheduling issues. But of course, real-world tasks may not start and end on the dates that MS-Project selects. This usually happen for two reasons:

- Often something needs to be done ahead of something else and therefore must be done first, possibly very much earlier than MS-Project chose. However this actually indicates a technical dependency between two tasks and all technical dependencies should be modelled in the plan using constraints. If a task can *only* be started when another is completed then they should be linked in that way. If this is done properly then MS-Project will *always* schedule the more important task first.
- The commonest reason for starting a task at a time other than when MS-Project has placed it on the plan is because the project team may arbitrarily choose to commence one task over another.

MS-Project does not force you to move a task on a plan just because you happen to have started at a different time. Even if you set some percentage complete or add values into the Actual Work field it does not work out that real work has been done ahead of schedule and move the task start date. So you actually end up completing work on tasks *in the future* as far as MS-Project is concerned!

You might not be too worried about this- after all if a task gets marked as complete do we care whether the plan thinks that is in the future or not? Well unfortunately MS-Project still tracks the resource usage so if you completed a task today that MS-Project had planned as taking all next week, then it will still expect the resource to be used up next week and so will not plan for other tasks in the same period, even though the task was completed ahead of schedule and therefore the resource is, notionally at any rate, free to do other things.

In section 4.3 (Enter explicit dependencies as constraints), it was advised that you do not constrain tasks in the plan so that they are forced to follow a set order or appear to start on the “right” data unless there was a need to do so because of a real dependency. Tasks should be left to float around in the plan starting and ending on the dates MS-Project chooses. But in order to get over the resource allocation problem above it becomes necessary at some point to ensure the task starts in the plan on the same day as it actually starts in the real world and it is easy to do without shackling the plan with constraints...

At the point that the task *actually begins* the project manager must alter the Start Date on the task to indicate to MS-Project that it has started!

This would typically be done at the point the Project Manager first records Actual Work against the task. The task moves on the MS-Project plan and work is recorded against it. MS-Project will not move it again unless you force it to. Since the task has moved, it may well have been put in parallel with other tasks and will also have left a “hole” wherever it was moved from, which may now be filled with work on other tasks. Simply Level the plan at this point and MS-Project will shuffle the outstanding tasks around to take these into account.

## 6.4 Putting Tracking into Practise

This block of activities, which I have called the Tracking Cycle, is the list of activities the Project Manager must undertake *in this order* to update the project and the Project Plan to reflect what is actually happening on a day-by-day or week-by-week basis. This should become second nature- whenever you need to understand whether your project is on-track you need to ensure the plan contains the most up to date position by exercising the tracking cycle. The plan will then tell you everything you need to know!

### 6.4.1 Gathering and Recording Actuals

The Project Manager should periodically (daily or weekly) gather information on what work has been started and completed and how much time has been spent on a task since the last time they asked.

The Project Manager should then go through the tasks on the plan and update the Actual Work and Remaining Work for the relevant tasks. Remember to set the Start Date field for the task when entering Actual Work for the first time, see section 6.3 (Task Start Dates). Always update them for an individual task in this order:

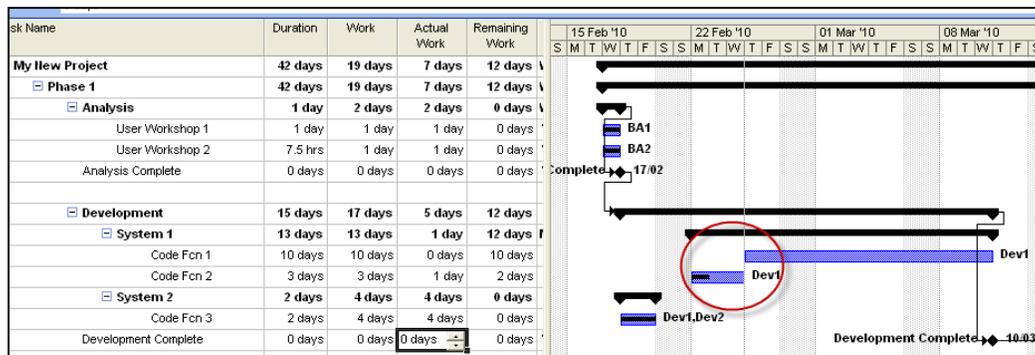
1. Actual Work (note- **always** add the new effort spent to the old value stored here)
2. Remaining Work

This will ensure the Work column is updated to show the new overall total estimated work for the task. Sometimes you will find, for tasks that are overrunning, that when you update the Actual Work column MS-Project immediately marks the task as complete because you have entered more work than was originally planned for the task (and so MS-Project calculates that there can be no remaining work, i.e. the task is complete). Don't panic! Merely move to the Remaining Work column and enter the effort that is estimated to be left remaining (if any). MS-Project will then re-open the task and recalculate the total Work column and the Percentage Complete value.

**Important Tip!**  
 Save a copy of the project plan in a separate place.  
 At this point **do not** overwrite the original!

### 6.4.2 Moving Unfinished Tasks

Let us imagine that the Project Manager has been tracking the example project and the date is now Thursday 25<sup>th</sup> February. They have gathered all the information and updated all the actuals but find the following issue:

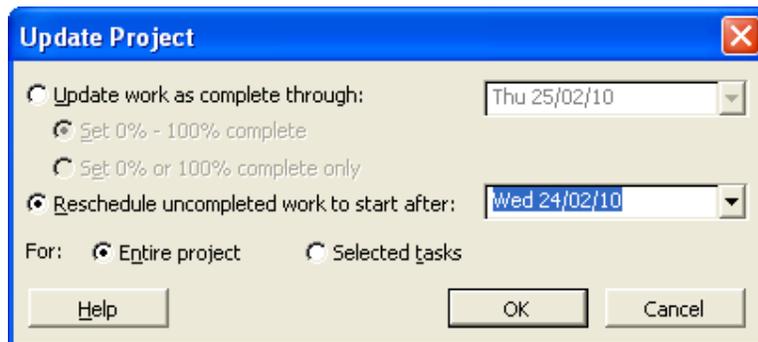


For some reason Dev1 has not made progress on Code Fcn 2. They were scheduled to do three days work on Monday, Tuesday and Wednesday, but were only able to work on the task for one day. They are still confident it is, overall, a three day task but the remaining two days

must now take place in the future and this will prevent them from starting the next task, Code Fcn1.

When the Project Manager has updated the actuals of all of the tasks that require it, the next steps is to move all unfinished tasks into the future. This is achieved using the Update Project panel.

*Tools => Tracking => Update Project*



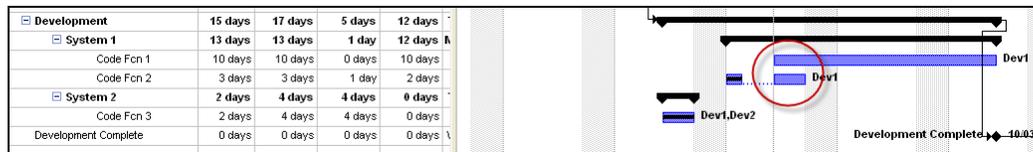
Here the Project Manager tells MS-Project to reschedule all uncompleted work to commence *after* a certain date. Since, according to our example, the date today is 25<sup>th</sup> February we want to schedule the work to commence *after 24<sup>th</sup> February*. Select the correct date and press the OK button.

MS-Project may, at this point, move all the tasks as required. Alternatively it may find that it cannot because to do so would break a constraint that has been set (for example it may try to move the start date on a task that has a “Must Start On” constraint which is why it is often better to use “Start No Earlier Than” to ensure tasks don’t start until a certain date) and MS-Project will display the following error.



When this happens it can become difficult to determine why MS-Project believes that it cannot reschedule certain tasks. It can often be helpful to add the column “Constraint Type” to your project as this allows you to see at-a-glance which tasks may be the culprits (tasks with the constraint type “As Soon As Possible” will almost certainly never be the cause of the problem). If all else fails you will need to manually update the start dates of tasks that got “left behind” in the past and set a constraint on them to “Start No Earlier Than” tomorrow.

Sometimes MS-Project will give this message for no apparent reason whatsoever, which is quite infuriating. However after rescheduling the unfinished tasks in the example the project plan now looks like this:

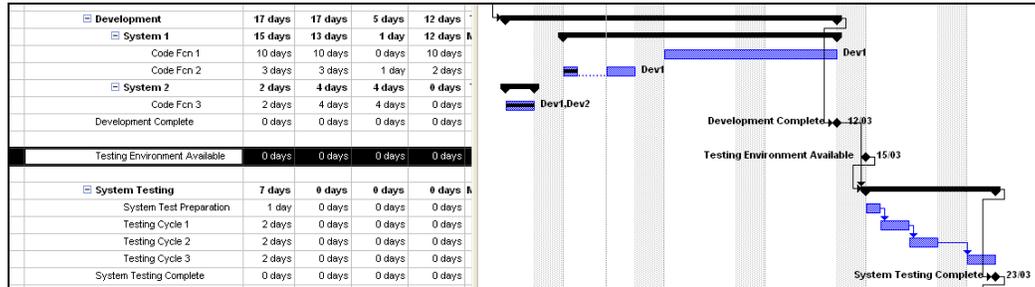


Note that MS-Project has moved the unfinished part of the Code Fcn2 task to commence from “today”. Unfortunately that work now clashes with the task that Dev1 was supposed to be working on from 25<sup>th</sup> Feb, Code Fcn1, however this will get sorted out when the plan is levelled.

**Important Tip!**  
 When you are happy that you have moved all unfinished work into the future, save *another* copy of the project plan separate from the original.  
 At this point **do not** overwrite the original!

### 6.4.3 Levelling the Plan

Having made various changes to the task actuals and moved uncompleted work around, the plan is likely to be showing overloading on resources as tasks have been moved in parallel with other tasks. As we saw in section 4.5 (Level the Plan) this means the plan will almost certainly be showing incorrect milestone dates because it currently relies on project personnel doing much more than one days work per working day. Follow the instructions in section 4.5 (Level the Plan) to level the plan and show the new milestone dates.



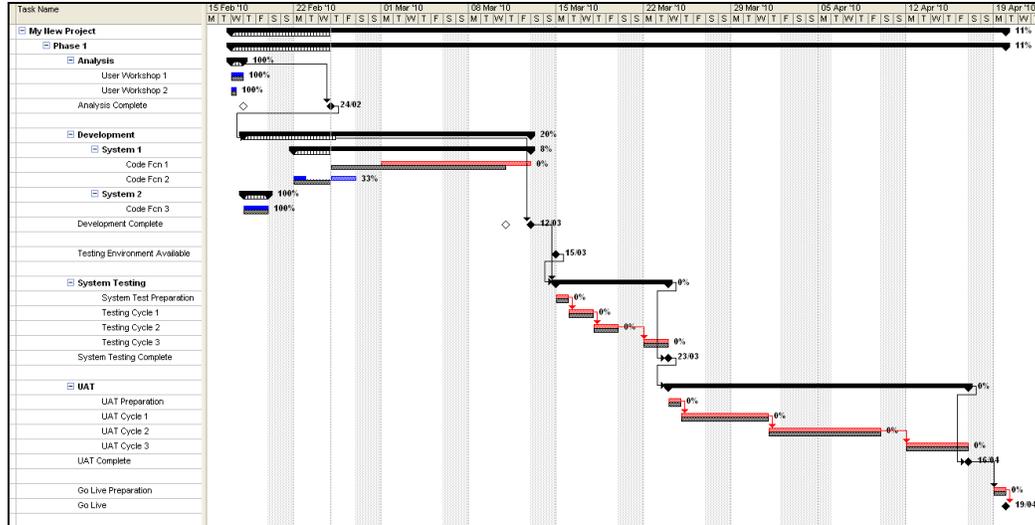
We can now see that Code Fcn1 has been moved to start later and allow Code Fcn2 to complete. This in turn has made the Development Complete milestone move to 12<sup>th</sup> February. Fortunately, in this case, there was some slack before testing started caused by the unavailability of the testing environment and that slack swallowed the overrun of the development. However it is now very easy for the Project Manager to see that any further delays in development will affect the start date of system testing.

Levelling the plan after changes to the actuals can sometime produce a project plan that is not consistent with the requirements of the project. To put it another way, the plan can sometimes go a little wild at this point! If this happens to you, don't worry, it happens to everyone and is one of the joys of using MS-Project! It is here that the Project Manager may want to move tasks around, change priorities to reflect the current position, bring in more resource or rebalance resources' workloads by reallocating. It is at this point you will realise the benefit of having saved separate copies of the plan and not overwritten the original! Sometimes it is easier to go back to the original and start again and sometimes MS-Project can twist a project so badly that your *only* course of action is to go back to the original and start again. Make sure you **never** overwrite the original when doing a tracking cycle until you are completely happy with the current plan. You have been warned!

### 6.4.4 The Tracking Gantt

One of the useful views of the project provided by MS-Project is the Tracking Gantt.

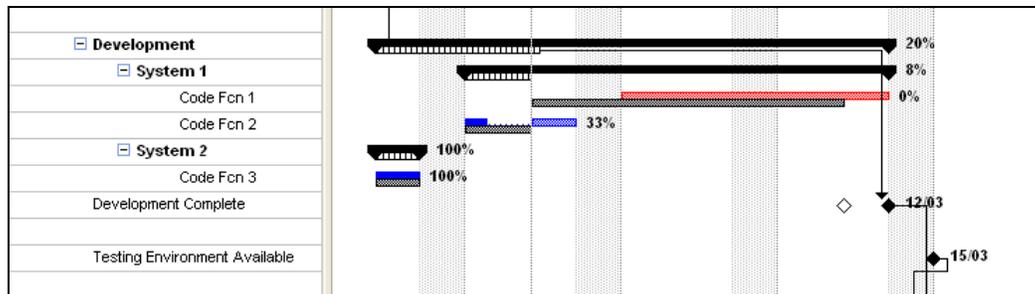
View => Tracking Gantt



This looks similar to the normal Gantt chart but it shows additional information (assuming that a Baseline has been saved).

Perhaps the most important information it yields is the Critical Path for the project- This is comprised of all the tasks with red shading. The critical path is the sequence of tasks that must be completed on schedule for the entire project to be completed on schedule. In other words if anything on the critical path slips it is unavoidable that the final delivery of the project will also slip.

On the tracking Gantt all tasks are shown with two horizontal bands instead of one.



The bottom band is the task where it appeared on the baseline project plan. The top task is where it appears on the current plan. The percentage on the right-hand side of each task is its Percentage Complete value.

When milestones slip they leave behind a diamond shape on the Tracking Gantt to show where they appeared on the baseline plan