Pro Visual Studio 2005 Team System

Jeff Levinson and David Nelson

Apress®

Pro Visual Studio 2005 Team System

ISBN-13 (pbk): 978-1-59059-460-5

ISBN-10 (pbk): 1-59059-460-6

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Ewan Buckingham
Technical Reviewers: Gautam Goenka, Bata Chadraa, Anutthara Bharadwaj, Munjal Doshi, Winnie Ng, Joe Rhode, Siddharth Bhatia, Amy Hagstrom, Yogita Manghnani, Tom Patton, Alan Hebert, Bill Essary, Sam Jarawan, John Lawrence, Jimmy Li, Bryan MacFarlane, Erik Gunvaldson, Adam Singer, Chuck Russell, Kathryn Edens, Patrick Tseng, Ramesh Rajagopal, John Stallo, Jochen Seemann, Michael Fanning, Ed Glas, Eric Lee, Bindia Hallauer, Michael Leworthy, Jason Anderson, Michael Koltachev, Boris Vidolov, James Su, Thomas Lewis, Steven Houglum, Bill Gibson, Ali Pasha, Dmitriy Nikonov, Prashant Sridharan
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade
Project Manager: Sofia Marchant
Copy Edit Manager: Nicole LeClerc
Copy Editors: Marilyn Smith, Jennifer Whipple
Assistant Production Director: Kari Brooks-Copony
Production Editor: Katie Stence
Compositor: Dina Quan
Proofreader: Nancy Riddiough
Indexer: Brenda Miller
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The source code for this book is available to readers at http://www.apress.com in the Source Code section. You will need to answer questions pertaining to this book in order to successfully download the code.

CHAPTER 4

Project Management

Neither of the authors has ever been a project manager. We have had the luxury of leading teams from a technical perspective and have left the high-profile leadership to others. Project management is, to say the least, a thankless job at times. We have worked with numerous project managers and have seen the problems they face. These problems range from getting developers to track their time against work items, to explaining to the customer why the project is not going as smoothly as the customer imagines it will go. There are many more problems in between that the project manager has to deal with.

At the base of the project manager's job is creating schedules, recording data, and reporting data. This is where Visual Studio Team System plays a role in simplifying the project manager's role. The tools in VSTS that relate to project management help bring together tasks performed by Microsoft Project (or Excel), work item tracking, and the Project Portal. VSTS allows the project manager to create schedules in Microsoft Project (these schedules include the work items for the team members), add work items from Microsoft Excel (or Project), track changes to work items via the project schedule, and publish updates to the Project Portal. No longer do team members have to report status to the project manager—the project manager can just pull the data.

Note Okay, let us face facts. Team members (and you know who you are) do not report anything to the project manager. It's like pulling teeth. So, in this case, project managers no longer have to pester team members to report what they are working on.

The heart of this system is the work item tracking system and the version control system. They work together to supply the information the project manager needs to provide accurate information in a timely fashion to the appropriate people—the people who ultimately continue to fund the development of a project.

In this chapter you will see how the integration with MS Project works, and how to add work items, update work items, and refresh work items using both Microsoft Project and Microsoft Excel. We discuss work items in great detail in Chapter 5.

A Week in the Life of a Project Manager (without VSTS)

Before showing you how to use VSTS to manage projects, let's take a look at what project managers actually do by looking at the starting week of a new project.

Day 1, Monday

The project manager, whom we'll call John, sets up Microsoft Project with all of the resources available at the time (typically, the system architect and/or the application architect, and possibly a lead developer and tester). John creates the basic outline of the project (system development life cycle phases and high-level deliverables) and assigns some deliverables to the available resources. In real life this usually takes more than a day but in this case John is very fast.

Day 2, Tuesday

John creates Excel spreadsheets for each member of the project team. He transfers the tasks they are responsible for from Microsoft Project to each of the spreadsheets (if John is really technical, he has written or has used a previously created macro to do this automatically). John has to do this at the beginning or end of every week for the following week for each member of the team. This becomes more tedious as more resources are added.

Day 3, Wednesday

Now the project manager has to set up a communication plan with the stakeholders. John has to find out what the stakeholders want to know and when they want to know it. He needs to figure out how to configure a change management system (i.e., create the process for the system and integrate it with the tools—usually Excel spreadsheets and e-mails). This is sometimes exceedingly difficult because different classes of stakeholders want to know different things.

Day 4, Thursday

On the fourth day of the week, the project manager has to set up shares, folders, etc., for the project team to store their files (user requirements, owner specifications, architecture documents, etc.). He then has to publicize and document the structure so the location of all of the deliverables and related documents is known.

Day 5, Friday

The end of the first week. Whew. John is almost done. Now he just needs to find out what everyone did for the week so he can report their status to the stakeholders and find out what is left to do for each deliverable assigned to the various project team members. The only problem is that no one filled in the Excel time sheets and mailed them to John. Now he has to send

e-mails and hound team members for their status. Once they send him the Excel spreadsheets (because everyone on the team is really good at doing that) he has to extract all of the numbers and update Microsoft Project with the number of hours left to complete a deliverable, or mark the deliverable as complete.

Two Days in the Life of a Project Manager (with VSTS)

Now, let's see that scenario with Visual Studio Team System's project management integration capabilities.

Day 1, Monday

The project manager creates a new team project using Visual Studio Team System (selecting the appropriate methodology template). He adds all of the resources that he knows about to the new team project. John opens Microsoft Project and adds all of the project resources to the new project file. John creates the basic outline of the project (SDLC phases and high-level deliverables) and assigns some deliverables to the available resources. In real life this usually takes more than a day, but in this case John is very fast. John updates the changes he made in Microsoft Project with the Team Foundation Server.

Day 2, Tuesday

John sends out an e-mail to the stakeholders detailing where the team project portal is and how to access the available reports. John sets the policy that changesets must be associated with work items, and then John spends the rest of the week fishing.

Okay, it is not quite that cushy for a project manager, but you get the idea. Much of the tedious work simply does not need to be done. In the rest of this chapter, you will see how this is actually accomplished.

■ **Note** There is a lot more that goes into creating a project schedule than is described here. For the purposes of this chapter, only those items that are directly related to Visual Studio Team System will be discussed. The "schedule" will only consist of the items assigned when the project was created.

Using Microsoft Project

The first thing a project manager will do (after the team project is created) is create a project schedule. The project schedule consists of a series of tasks, which may also include other tasks—in other words, a hierarchical structure of tasks. They consist of start times, due dates, descriptions, and to whom the tasks are assigned.

■ **Note** Tasks are a type of work item. These terms are used interchangeably in this chapter.

To begin with, launch Microsoft Project from the Start menu (Start ➤ Programs ➤ Microsoft Office ➤ Microsoft Project). If everything is installed correctly, you will see an additional toolbar in Project (shown in Figure 4-1).

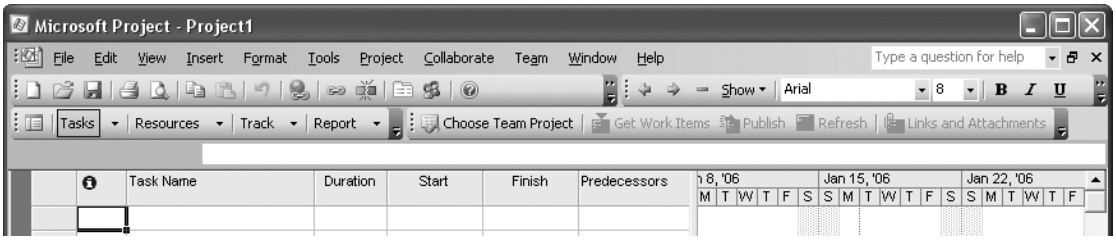


Figure 4-1. *VSTS toolbar in Microsoft Project*

As you can see, before you can create a schedule, you need to select the team project this schedule will be associated with. To do this, select Choose Team Project (also available from the Team menu on the main menu). This brings up the Connect to Team Foundation Server dialog box shown in Figure 4-2.

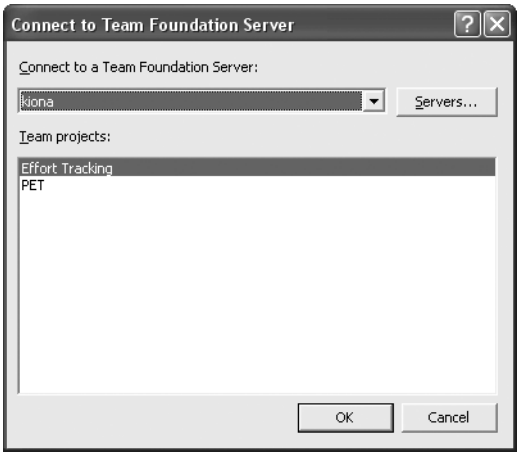


Figure 4-2. *Connect to Team Foundation Server dialog box*

Select the server where your project resides, then select the Effort Tracking project and click OK.

■ **TIP** You can also open a project from Team Explorer by selecting Server ➤ [Project Name] ➤ Project Management ➤ Development Project Plan.mpp which is associated with the project by default.

Once you have done this, the Get Work Items, Publish, and Refresh options become available to you. At this point though you still do not have any work items visible.

Retrieving, Adding, and Updating Work Items

To get the work items from the server, select Get Work Items from the toolbar. This will display the Get Work Items dialog box shown in Figure 4-3.

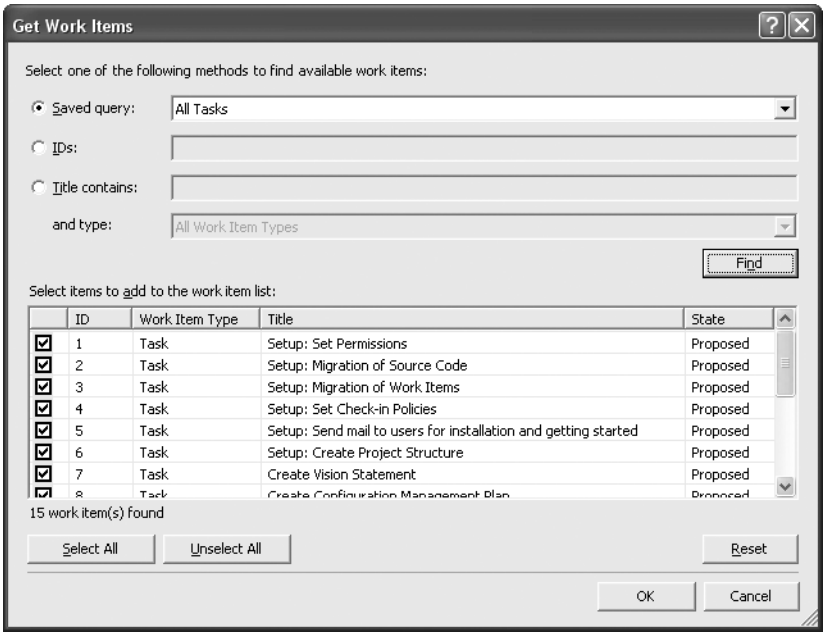
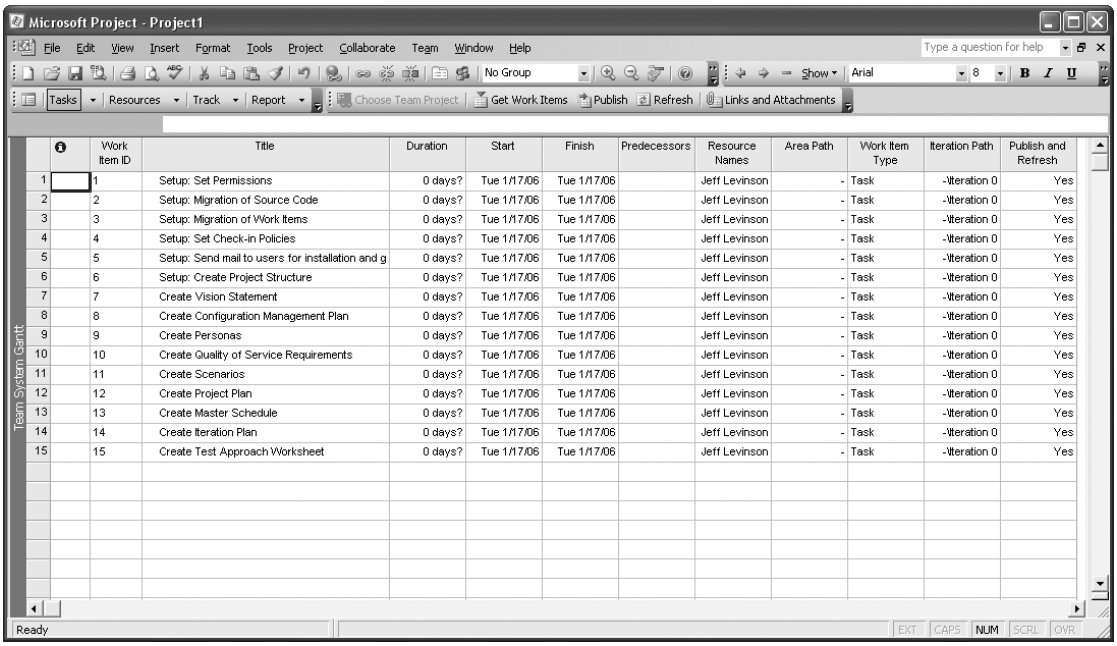


Figure 4-3. *Get Work Items dialog box*

For this exercise, select the All Tasks saved query and click Find. Note though that you can search on various items depending on what your needs are. At the beginning of a project, the All Tasks query is a good place to start, as these are the only tasks that are currently part of the project. By default, everything returned by the lookup is selected for inclusion in the project schedule. Click OK to import the items into Microsoft Project (Figure 4-4).



	Work Item ID	Title	Duration	Start	Finish	Predecessors	Resource Names	Area Path	Work Item Type	Iteration Path	Publish and Refresh
1	1	Setup: Set Permissions	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
2	2	Setup: Migration of Source Code	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
3	3	Setup: Migration of Work Items	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
4	4	Setup: Set Check-in Policies	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
5	5	Setup: Send mail to users for installation and g	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
6	6	Setup: Create Project Structure	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
7	7	Create Vision Statement	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
8	8	Create Configuration Management Plan	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
9	9	Create Personas	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
10	10	Create Quality of Service Requirements	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
11	11	Create Scenarios	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
12	12	Create Project Plan	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
13	13	Create Master Schedule	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
14	14	Create Iteration Plan	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes
15	15	Create Test Approach Worksheet	0 days?	Tue 1/17/06	Tue 1/17/06		Jeff Levinson	-	Task	-\Iteration 0	Yes

Figure 4-4. *Initial project schedule*

There are a couple of small items to note here:

- The start and end times for all tasks are for the day the project is created and have a duration of zero days.
- No predecessors are set.
- The resource for all items is the individual who created the team project.
- There are no project areas at this point, so no changes can be made there.
- All iterations are iteration 0, even though there are three iterations, since it is assumed these tasks occur at the start of the project.

At this point you can change whatever you want to change as needed. To begin with, you would probably remove any items that were not needed on your project. To delete an item, select it (item 15 is a good choice here) and press the Delete key (or right-click the item and select Delete. After you have completed making changes, you must publish the changes to the server. To publish the changes, just select the Publish button from the toolbar.

■ **Tip** You can elect *not* to publish an item by selecting either No or Refresh Only from the Publish and Refresh column. Selecting No indicates that the item will never be published, and selecting Refresh Only indicates that the item will be refreshed, but no changes you make to the item will ever be published.

This updates Team Foundation Server and if you added any items will return the Work Item ID to populate the project schedule.

■ **Caution** In version 1 of VSTS you cannot assign multiple resources to a single task. You have to duplicate the task and set one resource per task. The VSTS development team is actively looking at this issue. If you try to assign multiple resources to a single task, an error message will be displayed when you publish the changes.

The ability to publish only certain items is important. A project manager would probably *not* want to publish the roll-up tasks because those are not tasks that can be "worked," per se.

■ **Tip** To assign work items to multiple resources, you must create the work item several times and assign only one resource to each work item.

Adding Attachments and Links

In addition to adding, retrieving, and updating items, you can also attach files and links to specific work items. To attach files and links, an item must already be published. Select a published item in the list and then select Links & Attachments. This will bring up the View/Edit Work Item Links and Attachments dialog box (Figure 4-5).

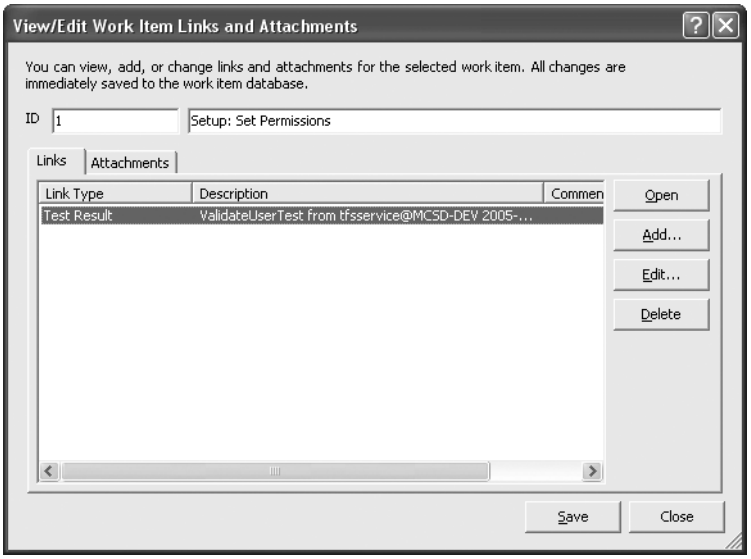


Figure 4-5. *View/Edit Work Item Links and Attachments dialog box*

A link can be one of the following: Changeset, Work Item, Versioned Item, Test Result, or a regular hyperlink. Depending on the type of link you are adding, the options are different. The Add Link dialog box is shown in Figure 4-6.

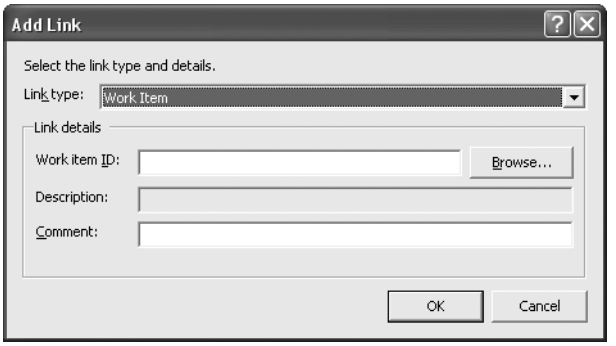


Figure 4-6. *Add Link dialog box*

The dialog in Figure 4-6 shows the options for a Work Item link type. Clicking the Browse button will display the Get Work Items dialog box shown in Figure 4-3. The Find Changesets dialog box will be displayed if you elect to link to a changeset (Figure 4-7).

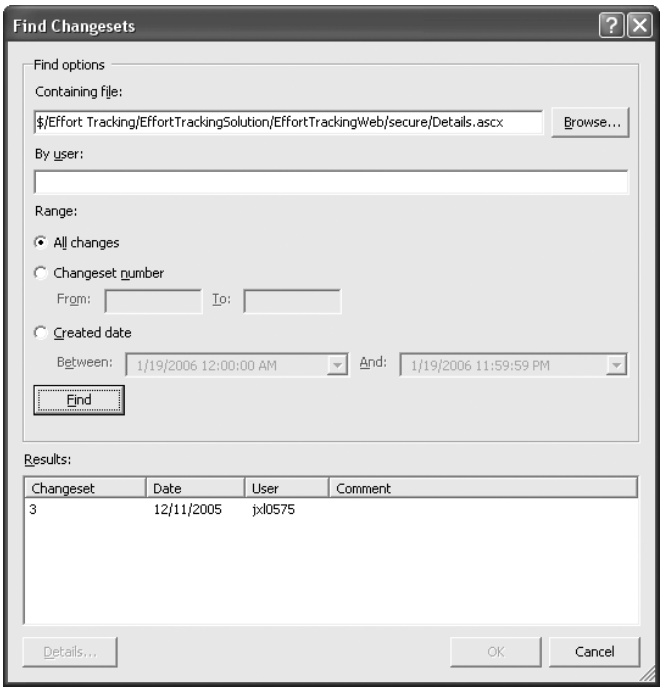


Figure 4-7. *Find Changesets dialog box*

■ **Note** While project managers *can* associate work items with changesets, it is generally not something a project manager would often do. Consider it as a tip for what you can do within Microsoft Project.

For the Find Changesets dialog, you browse to a specific file (contained in the source code control system) or enter a specific user's name and click Find. This will return all of the changesets for either that file or that user. The Range options let you filter the result set more granularly. From there you may either view the details of the changeset or select a result to link to.

The Versioned Item option provides the same dialog, only it lets you link to the item in a specific changeset (which you can browse for) or to the latest version of the item.

Selecting the Test Result option lets you browse for a specific test result (Figure 4-8).

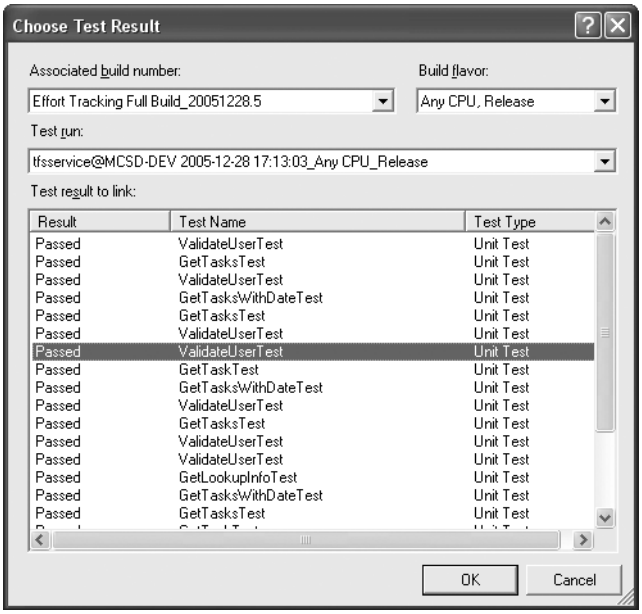


Figure 4-8. *Choose Test Result dialog box*

In this dialog box you can select the build number, the type, and a specific test run. From there you can select the specific test to link an item to. In this case, select any test that appears in the list and click OK.

■ **Tip** For the project manager, many of these options are a way of tracking developers' time and ensuring that a given task is completed. Using the Test Result link as an example, if a bug in the system needs to be fixed, a bug work item would be created (work items are discussed in depth in Chapter 5). When fixed, the developer could mark it as completed, and a project manager (or a developer, or a tester) could associate the results that validate the fix with the bug.

All link types allow you to enter a comment against the link.

Once you have selected your links, they will be displayed in the list view in Figure 4-5. Selecting an item in the list and then clicking Open will cause various windows to open, depending on the type of link you select. With the previous example, using a test result link, clicking the Open button displays the result shown in Figure 4-9.

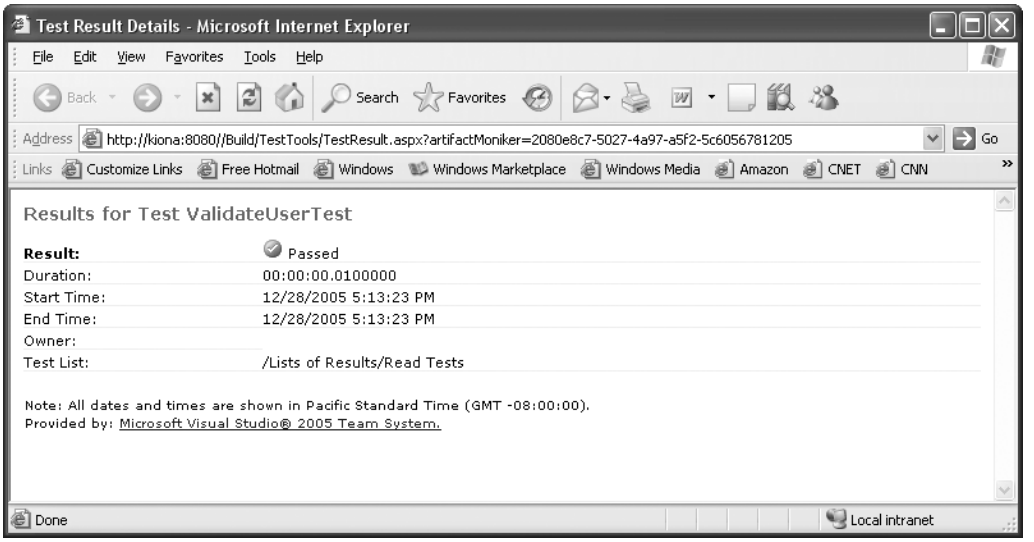


Figure 4-9. *Test Result Details link*

Adding an attachment is fairly straightforward. Click the Attachment tab, click Add, browse to the file you want to attach, and select OK. You can also enter a comment against the file. The one difference here is that links are automatically saved to the server as they are entered; files are not. When you have finished attaching all of the files you want, click the Save button. This will upload the files to the server.

Areas and Iterations

Areas and iterations can be added, deleted, and configured from within Microsoft Project, which displays the same dialogs as described in detail in Chapter 2.

Column Mapping

Finally, a project manager can view column mappings in Microsoft Project. Figure 4-10 shows a column displayed in Microsoft Project and the equivalent column in the VSTS database. You cannot actually make changes to the mapping through this dialog box. In order to make changes you will need to use the TFSFieldMapping command-line tool described next.

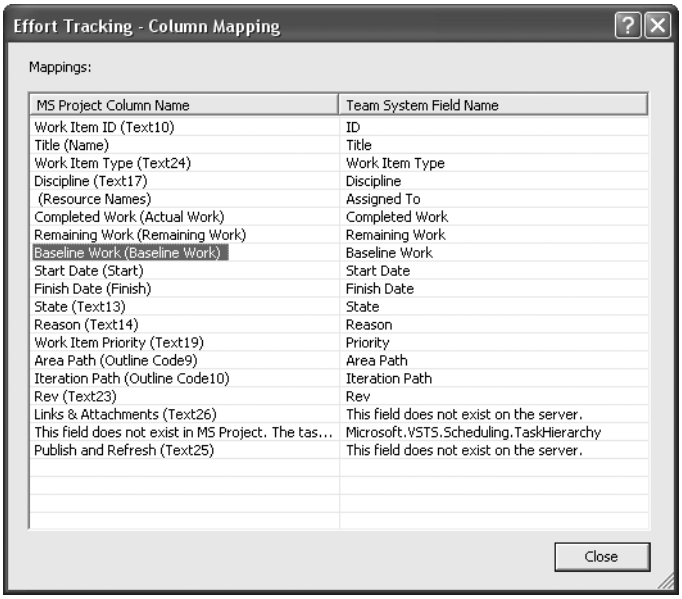


Figure 4-10. *Column Mapping dialog box*

■ **Note** The work item tracking command-line tools provide the ability to add and remove columns from the TFS database in order to support custom fields. These tools are discussed in Chapter 5.

To use the TFSFieldMapping tool, open the Visual Studio Command Prompt. Before changing mappings, you need to have a copy of the mapping file. To download the mapping file, run the following at the command prompt (or replace the items with appropriate values):

```
Tfsfieldmapping download kiona "Effort Tracking" "c:\mapping.xml"
```

This will download the mapping file for your project. The file is in, as with almost every other file in VSTS, an XML format. This file is shown in Listing 4-1.

Listing 4-1. *TFS Field Mapping File*

```
<?xml version="1.0" encoding="utf-8"?>
<MSProject>
  <Mappings>
    <Mapping WorkItemTrackingFieldReferenceName="System.Id"
      ProjectField="pjTaskText10" ProjectName="Work Item ID"/>
        <Mapping WorkItemTrackingFieldReferenceName="System.Title"
          ProjectField="pjTaskName" />
     <Mapping WorkItemTrackingFieldReferenceName="System.WorkItemType"
       ProjectField="pjTaskText24" />
```

```
        <Mapping WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Common.Discipline"
          ProjectField="pjTaskText17" />
        <Mapping WorkItemTrackingFieldReferenceName="System.AssignedTo"
          ProjectField="pjTaskResourceNames" />
        <Mapping WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Scheduling.CompletedWork" ProjectField="pjTaskActualWork"
          ProjectUnits="pjHour"/>
        <Mapping WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Scheduling.RemainingWork"
          ProjectField="pjTaskRemainingWork"
          ProjectUnits="pjHour"/>
        <Mapping WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Scheduling.BaselineWork" ProjectField="pjTaskBaselineWork"
          ProjectUnits="pjHour"/>
        <Mapping WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Scheduling.StartDate" ProjectField="pjTaskStart"
          PublishOnly="true"/>
        <Mapping WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Scheduling.FinishDate" ProjectField="pjTaskFinish"
          PublishOnly="true"/>
        <Mapping WorkItemTrackingFieldReferenceName="System.State"
          ProjectField="pjTaskText13" />
        <Mapping WorkItemTrackingFieldReferenceName="System.Reason"
          ProjectField="pjTaskText14" />
        <Mapping WorkItemTrackingFieldReferenceName="Microsoft.VSTS.Common.Priority"
          ProjectField="pjTaskText19" ProjectName="Work Item Priority" />
        <Mapping WorkItemTrackingFieldReferenceName="System.AreaPath"
          ProjectField="pjTaskOutlineCode9" />
        <Mapping WorkItemTrackingFieldReferenceName="System.IterationPath"
          ProjectField="pjTaskOutlineCode10" />
        <Mapping WorkItemTrackingFieldReferenceName="System.Rev"
          ProjectField="pjTaskText23" />
        <ContextField WorkItemTrackingFieldReferenceName=
          "Microsoft.VSTS.Scheduling.TaskHierarchy"/>
        <LinksField   ProjectField="pjTaskText26" />
        <SyncField   ProjectField="pjTaskText25" />
  </Mappings>
</MSProject>
```

In order to change the mappings, simply update the WorkItemTrackingFieldReference-Name values or the ProjectField values. When you have finished making changes, upload this file back to TFS to complete the mapping update. To do that, use the same command as you did to download the file—just use the word “upload” instead of “download” and you are done.

■ **Tip** The PublishOnly setting is an incredibly useful field. As a project manager, you may not want people to be able to change start and end dates of work items via Microsoft Project or Excel. You may not want them to change other items that you consider important. Adding this attribute allows people to retrieve the value, but not update it.

Using Microsoft Excel

Microsoft Excel is everyone's favorite friend. If some major problem caused Excel to go down all over the world, virtually every business on the planet would grind to a halt rather suddenly. Knowing how useful Excel is to everyone, the VSTS team used the capabilities of Excel to allow team members to enter items into TFS through the Excel interface.

■ **Note** Excel is a part of Microsoft Office; Microsoft Project requires a separate license. In general, only the project manager needs to be using Project, and everyone else can use Excel to make entries if they need to (all entries by the development team can be made from within Visual Studio though).

If you are a project manager reading this, how often have you had to create macros to pull data from MS Project into Excel so developers can log their hours against specific items? If you are a developer reading this, how many times have you gone from project to project, where everyone has a different way of recording hours? The beauty of this solution is that it is all automatic. You can get your work items (if you need to fill this sheet in); you can get everyone's work items (if you are a project manager); you can customize the query to see if work was completed (or just filter this list); or you can perform a hundred other operations to slice and dice the data any way you see fit to gather useful information.

■ **Tip** Another key benefit of Excel, and one of the reasons Microsoft chose to use it as the interface, is its easy ability to create charts from any type of data in VSTS. Even though you have the benefits of SQL Server Reporting Services, using Excel is often much faster—especially if the reports are for the project manager only.

The Excel interface is slightly different from the Project interface. When you first launch a new Excel file, the toolbar shown in Figure 4-11 is displayed.

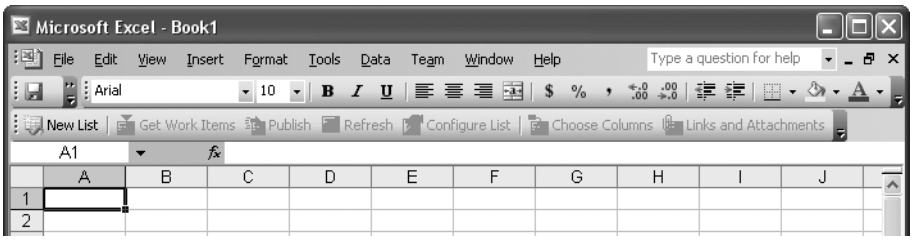


Figure 4-11. *The Excel Team System toolbar*

Creating Lists in Excel

Before you can do anything with Excel and VSTS, you need to create a new list by selecting New List from the toolbar. This displays the Connect to a Team Foundation Server dialog box (Figure 4-2). Once you connect to a Team Foundation Server you have the option of retrieving the results of a query or inputting new items (Figure 4-12).

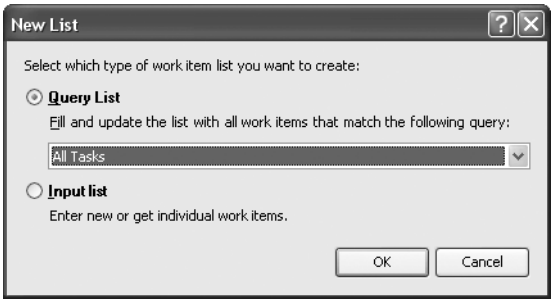


Figure 4-12. *The Excel New List dialog box*

For this first example, assume that you select the All Tasks query and click OK. This creates the list in Excel as shown in Figure 4-13.

This list should immediately validate why this is a popular tool for both project managers and other team members alike.

■ **Note** Some fields in this list are read-only. For instance, you cannot change the work item type once it has been created.

Publishing items from Excel works the same way as it does in Project—you just click the Publish button. However, you can also configure a list in Excel and choose which columns you want to view as part of the list.

Selecting the Input List option (Figure 4-12) creates a blank list with several default columns. This allows you to enter items without having to first retrieve items from the server.

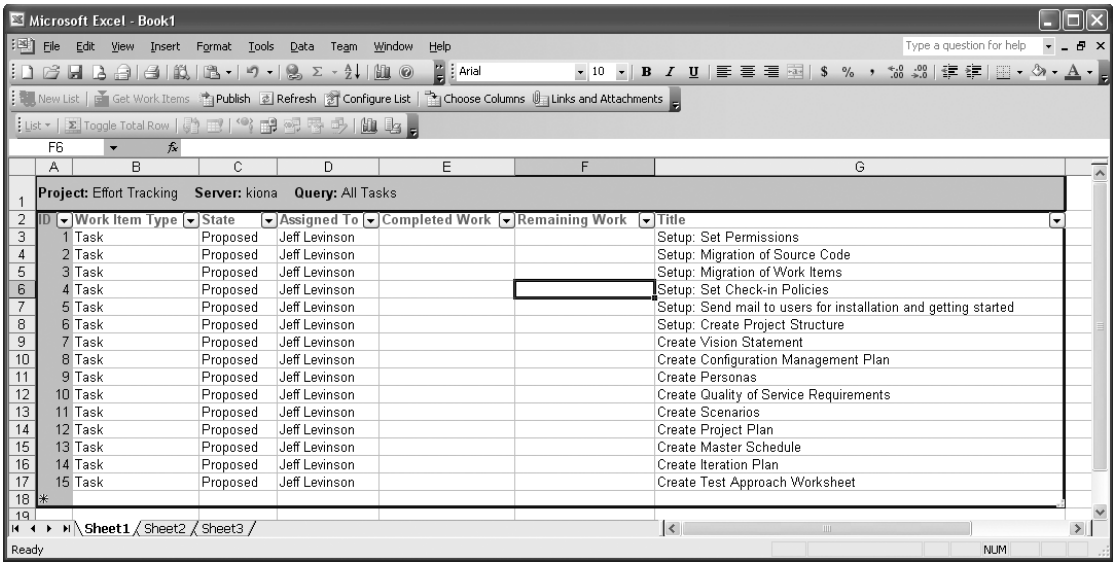


Figure 4-13. *Query results in Microsoft Excel*

Configuring Lists in Excel

To configure the list, select Configure List from the toolbar. This will display the Configure List Properties dialog box shown in Figure 4-14.

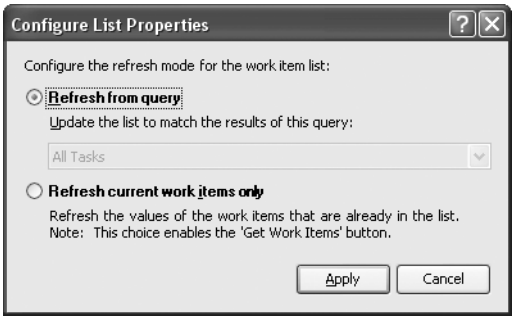


Figure 4-14. *Configure List Properties*

This dialog box in itself does not do anything; however, it affects how the list is updated from the server. The default is to Refresh from query. This means that whenever you click Refresh on the toolbar, the same query will be run over again and any new or updated items will be shown in the list. Selecting the "Refresh current work items only" option means that when you click Refresh, only items that are currently in the list will be updated.

■ **Note** Clicking Refresh also enables the Get Work Items button, which allows you to add other items not in the current list (see Figure 4-3 and the associated explanation).

The other option to configure lists in Excel is to select the columns you want to display in the list. To do this, select the Choose Columns button. This will display the Choose Columns dialog box shown in Figure 4-15.

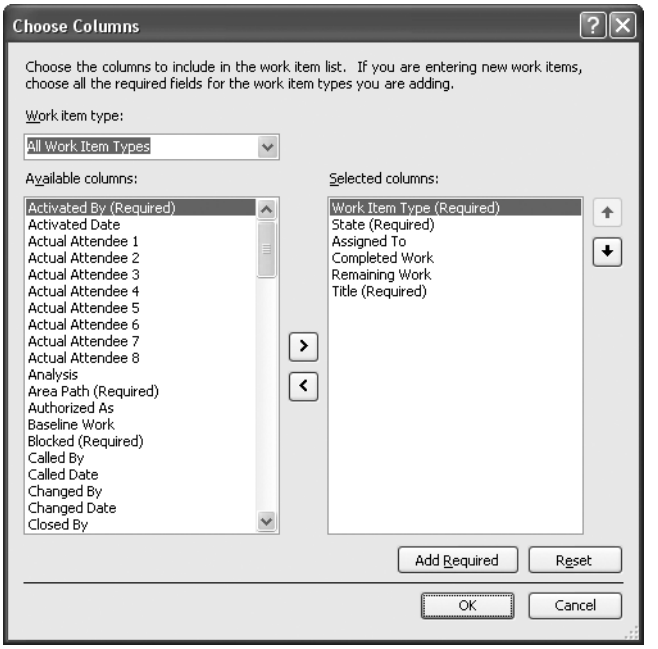


Figure 4-15. *Choose Columns dialog box*

This dialog box lets you choose columns associated with any work item type (via the drop-down list at the top of the form) because the list in Excel can display mixed item types. This means that the possible configurations in Excel are virtually endless depending on your need. The available columns are too extensive to list here, but any column you have seen so far, and in the next chapter, is available through this dialog box. To add a column, just select the column on the right and click the right arrow button. Click the left arrow button to remove an item from the selected columns list.

Clicking reset will reset the columns to the default for the given list. Selecting Add Required will add all required items from the Available Columns list (those identified with a "Required" after the column name) to the Selected Columns column.

Links and attachments work the same way as described in the "Adding Attachments and Links" section.

Using Visual Studio

You can of course use Visual Studio to perform project management duties. You can add tasks, assign tasks, and set start and end dates all from within Visual Studio (see Figure 4-16).

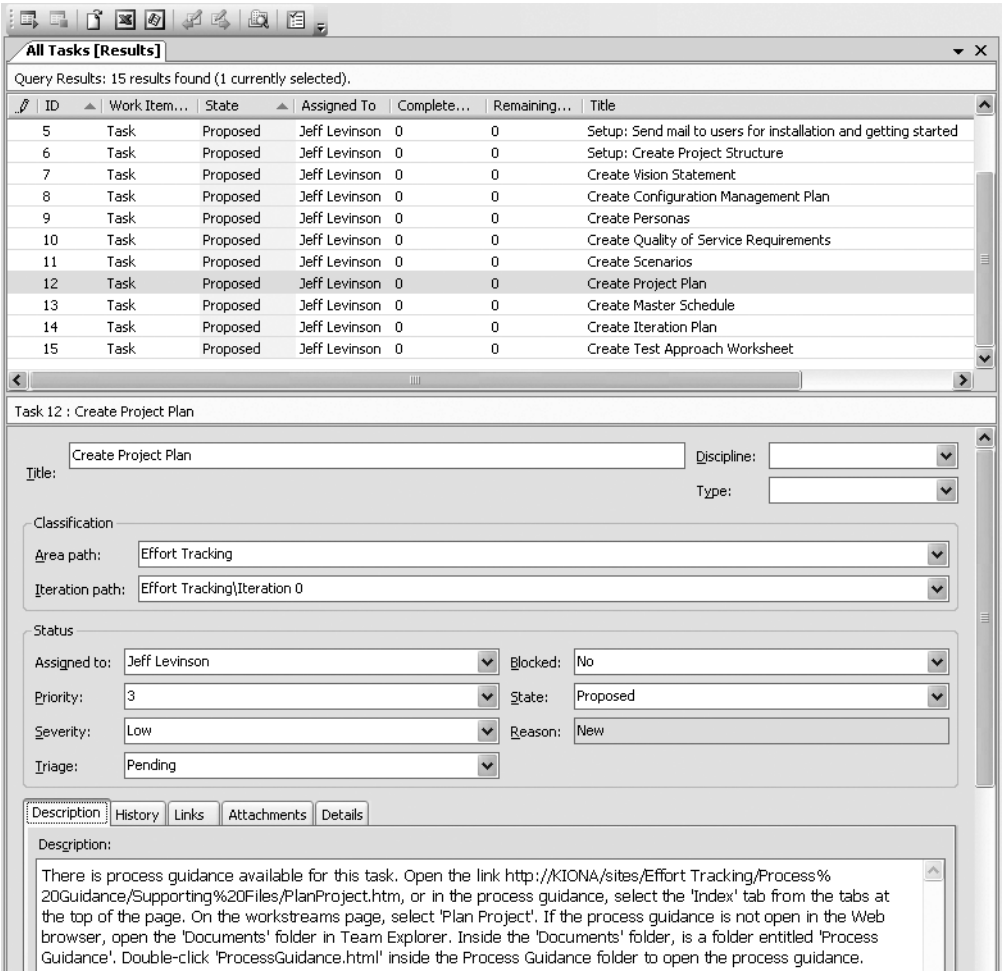


Figure 4-16. *Task list in Visual Studio*

The details tab contains the schedule information. Virtually everything else on this work item form (bottom portion of Figure 4-16) maps to a field in Excel or Microsoft Project. In general, this is not a very friendly view for project managers, which Microsoft realizes. This is the reason for the Excel and Microsoft Project integration. However, when a project manager finds himself or herself looking at a task list in Visual Studio (come to think of it, this applies to anyone who works with VSTS), the team member can view everything in the task list in Excel or Microsoft Project. To do this, simply select all of the items you want to export, and select the appropriate icon on the task list toolbar. Enough said about project management from within Visual Studio.

Summary

In this chapter you saw how the integration between Visual Studio Team System and Microsoft Project and Excel will save you countless hours of work and allow you to manage a project's data more easily. A project manager typically manages multiple software projects at once, because no one software project should take up all of their time. At least, that is the theory. The reality is that project managers often spend way too much time figuring out how they are going to record the team's time, and against which items, track the status of items, and report status to the stakeholders.

With the simple but powerful and well thought out integration described in this chapter, much of the grunt work is simply removed. Teams will now have a repeatable, simple way to track their activities and the hours logged against items, the amount of work left, whether estimates are accurate, and so on. It all leads to one thing: through continued use of the tool, teams will become more efficient and spend more time engineering software rather than performing administrative tasks (which no one likes anyway). The next chapter introduces you to the details of the work items.