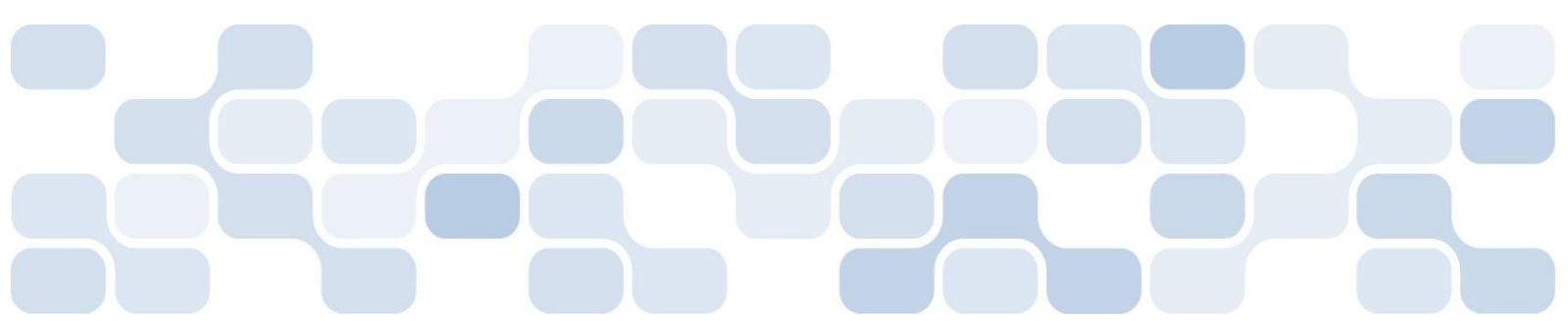


Managing Team Workflow with Visual Studio Team System 2008

White Paper

May 2008

For the latest information, please see www.microsoft.com/teamsystem



This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Excel, Project, and Visual Studio are trademarks of the Microsoft group of companies

All other trademarks are property of their respective owners.

CONTENTS

Introduction.....	1
Managing Work with Work Items	2
Work Item Creation.....	4
Default Work Items	11
Work Item Customization	15
Visibility and Traceability	19
Traceability.....	23
Conclusion	30
About the Author	31

INTRODUCTION

In an increasingly competitive market with pressures such as increasing complexity and shorter deadlines, companies need to manage their software development projects better. Project managers need more timely and accurate information regarding the work being done as well as the ability to dynamically change priorities.

Project managers need the best tools to:

- Ensure they prioritize work correctly and communicate priorities accurately to each team member.
- Communicate changes to work including task details, priorities, and task assignments.
- Help ensure only scheduled work is being done, thereby reducing wasted effort.
- Produce timely reports for business stakeholders.

Microsoft® Visual Studio® Team System 2008 offers many benefits to all members involved in the application life cycle from project managers to developers and testers. Team System is an integrated application life-cycle management solution suitable for teams of three or more. Following the success of Microsoft Visual Studio 2005 Team System, Microsoft has refined and enhanced the product set to offer even more value to your organization.

Value-up Versus Work-down

While this white paper does not focus on the emerging paradigm shift to value-up software processes, it is important to acknowledge that Team System provides an ideal foundation for realizing the benefits offered by the value-up approach. For more information on using the value-up approach with Team System, read *Software Engineering with Microsoft Visual Studio Team System* by Sam Guckenheimer (Addison-Wesley, 2006, ISBN 03210278720).

MANAGING WORK WITH WORK ITEMS

The efficient tracking and management of work is a significant challenge facing most software development teams today. Not only do they need to define and assign new work to a resource to complete, but they must also track, manage, and report defects in a convenient and timely manner. Visual Studio Team System 2008 Team Foundation Server provides a convenient and customizable work item tracking capability that enables teams to better manage, work, and track software defects than ever before.

A work item, as the name implies, is basically any unit of work that a team needs to track and complete during a software development project. A single work item may be one of a number of work item types including, but not limited to: task, bug, risk, and more. It is also possible to create your own work item types to help you to better capture additional fields for specific work items. This paper will demonstrate work item type customization in a later section.

Default Work Item Types

When creating a new team project, the Project Creation Wizard prompts you to select a process template. Among other things, this choice determines the default work item types as well as the initial work items that are created. The two out-of-the-box process templates provide different default work item types, as shown in Table 1:

MSF for Agile Software Development	MSF for CMMI Process Improvement
Task	Task
Bug	Bug
Risk	Risk
Quality of Service Requirement	Requirement
Scenario	Review
	Change Request
	Issue

Table 1: Default work item types.

Each work item type has specific fields, values and states relating to its purpose. Table 2 lists a few examples of the specific fields in a sample of work item types:

Work Item Type	Examples of Type-specific Fields
Bug	Triage (Approve, Investigate)
Risk	Severity (Low, Medium, High, Critical)
Change Request	Justification, Analysis

Table 2: Examples of work item type specific fields.

Internally, Microsoft uses a number of custom work item types including experience, test issue, feature, value proposition, and patent to name a few. You should consider your own environment and determine what custom work item types would offer benefits to your team.

WORK ITEM CREATION

Central to managing team workflow with Team System is the requirement to ensure all work is captured in a work item. This is important for not only ensuring the work gets completed but also for ensuring the accuracy of many of the automated reports in Team System.

Fortunately Microsoft has made it easy to create work items using a number of applications you are undoubtedly already familiar with in the Microsoft Office suite. This section examines the efficient creation of work items using Team Explorer, Microsoft Excel, Microsoft Project, Team System Web Access, and finally, through the use of the Team Foundation Server API. Each of these tools offers something a little bit different for each member of the project team.

The team member or members that create work items vary from team to team. Project managers may create high-level work items while technical team leaders may refine these and create the more granular work items that they assign to each team member. By default, all members of the Project Contributors and Project Administrators groups can create new work items.

Using Team Explorer to Create Work Items

Microsoft Visual Studio Team System 2008 Team Explorer is an add-in that is hosted inside the Visual Studio 2008 shell and is the most common interface for team members to use when working with work items.

All team members with a Team Foundation Server Client Access License (CAL) are entitled to install the Team Explorer and use it to interact with Team Foundation Server. For developers and team members that generally spend most of their time in the Visual Studio IDE, this will be the most convenient location to create and manage work items. The following steps show you just how easy it is to get started creating a new work item. For the purpose of this walkthrough, I will use a new team project named ABDemo, which was created using the MSF for Agile Software Development process template.

1. Start Visual Studio 2008 and make sure you have connected to your Team Foundation Server.
2. Expand the node for the team project in which you want to create a new work item.
3. Right-click the "Work Items" node, select "Add Work Item" from the context menu, and then select "Task" as the work item type (see figure 1).
4. You can then complete the fields in the Task work item and click Save to commit the new work item to the Team Foundation Server.

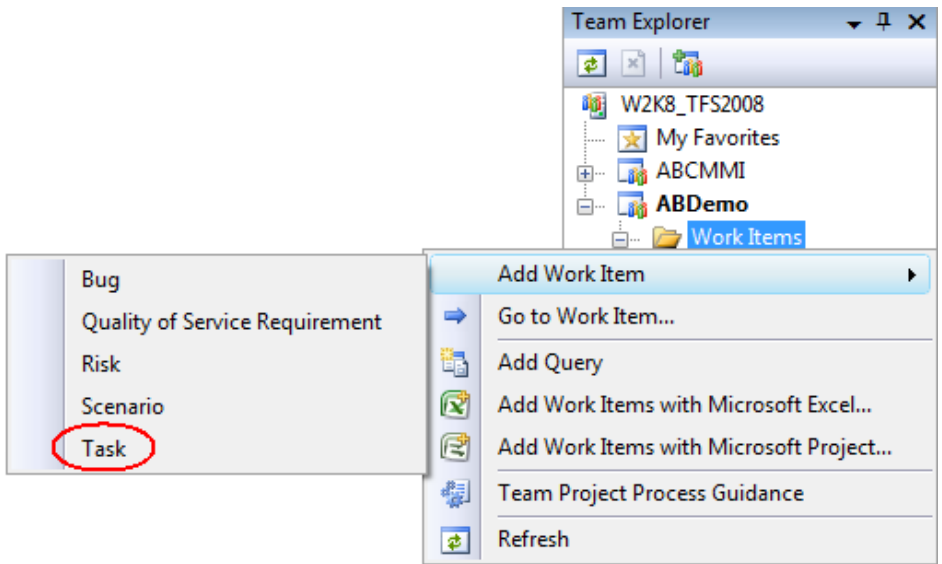


Figure 1: Creating a task with Team Explorer.

Using Microsoft Excel to Create Work Items

Installing Team Explorer provides a number of extensions to Microsoft Excel that enables you to create and manage your work items directly in Excel. Through integration with Excel, you can create work items in a worksheet and publish these back to Team Foundation Server with the click of a button.

Excel is an ideal choice for creating large numbers of work items or making changes to multiple work items simultaneously. For example, using the capabilities native to Excel like “Fill-Down”, you could reassign large numbers of work items to an alternative resource quickly and easily.

Figure 2 shows the familiar Excel interface with the Team System add-in that enables you to interact with your Team Foundation Server.

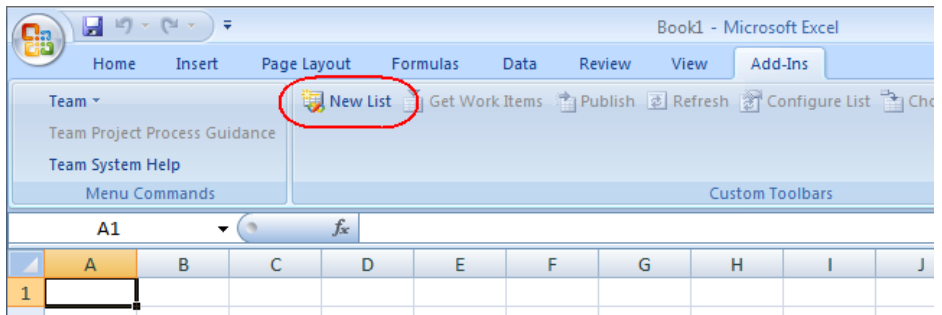
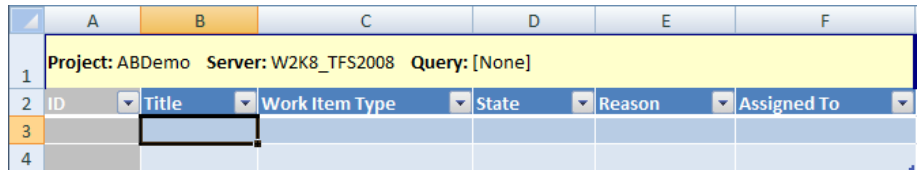


Figure 2: Working with Team Foundation Server work items in Microsoft Excel.

1. In the New List dialog box you have the opportunity to retrieve existing work items by selecting a query (for editing) or you can select Input List to add new work items from scratch. For now, select Input List and click OK.
2. Your worksheet will then display the six default fields where you can start entering information into five of them. (You can't enter the ID

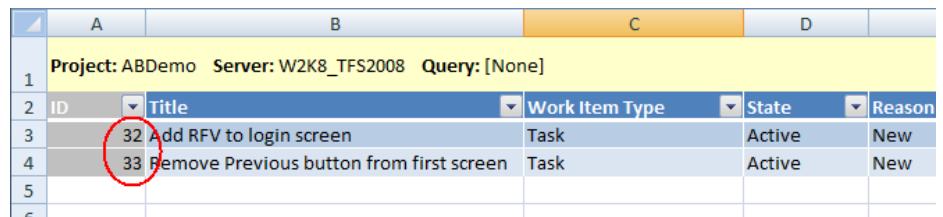
manually—it is added when you publish the work items back to Team Foundation Server.)



	A	B	C	D	E	F
1	Project: ABDemo Server: W2K8_TFS2008 Query: [None]					
2	ID	Title	Work Item Type	State	Reason	Assigned To
3						
4						

Figure 3: Creating a new work item in Microsoft Excel.

4. You can easily add additional fields to the worksheet by returning to the Add-Ins menu and clicking the Choose Columns button. You can then add and re-order the fields.
5. Add the new work items, remembering to take advantage of the features of Excel such as Fill-Down (CTRL+D), to make creating multiple similar work items easier.
6. Once you have entered the new work items, you need to commit these to the Team Foundation Server by clicking the Publish button from the Add-Ins menu. Any validation issues will be highlighted. You will need to resolve these conflicts and publish to Team Foundation Server again.
7. Once Excel has stored the new work items in the Team Foundation Server Work Item Tracking database, you should notice the Work Item ID fields will now display a number.



	A	B	C	D	
1	Project: ABDemo Server: W2K8_TFS2008 Query: [None]				
2	ID	Title	Work Item Type	State	Reason
3	32	Add RFV to login screen	Task	Active	New
4	33	Remove Previous button from first screen	Task	Active	New
5					
6					

Figure 4: Work Item IDs are assigned once you publish to Team Foundation Server.

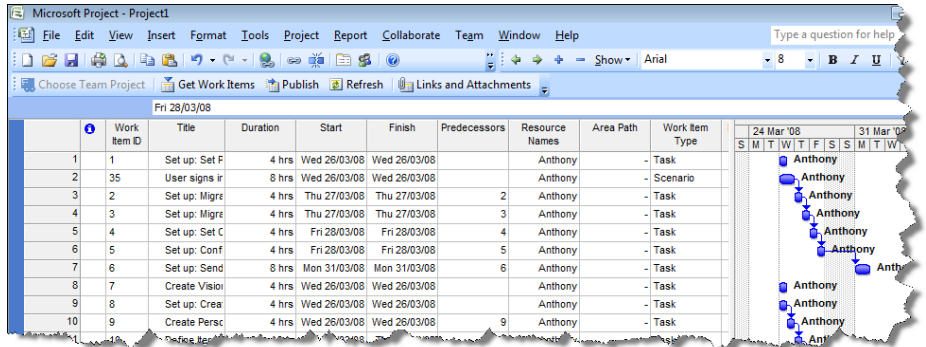
8. You can now close Excel, generally choosing not to save the workbook because the data is safely stored in Team Foundation Server.

In summary, Microsoft Excel provides a familiar and convenient interface for creating and editing multiple work items.

Using Microsoft Project to Create Work Items

Project managers will most likely prefer to use Microsoft Project to create and manage work items for their teams. Once you have installed Team Explorer on a computer with Microsoft Project, you can use the extensions to work with work items in much the same way as Microsoft Excel.

One of the unique benefits of using Microsoft Project to edit work items is the ability to schedule tasks and visualize project schedules using a Gantt chart.



The Gantt chart view helps you manage your team's workflow by ensuring your resources are effectively utilized.

Figure 5: Microsoft Project.

Click the "Choose Team Project" button on the toolbar.

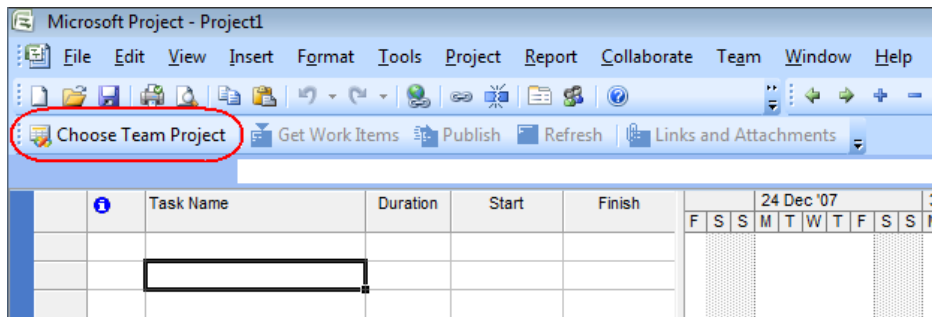


Figure 6 provides: Choosing your team project in Microsoft Project.

Select your Team Foundation Server, adding it if necessary, and select the perfect interface for Team Foundation Server Team Foundation Server Team Foundation Server Team Foundation Server scheduling work items.

Using Microsoft Team System Web Access to Create Work Items

The Visual Studio Team System Web Access 2008 Power Tool (TSWA) is a great Web-based UI for Team Foundation Server. Once you've installed it onto your server, users with a Client Access License (CAL) can start using this AJAX-enabled Web application. Using Team System Web Access, users can create and manage their work items, execute work item queries, view reports, and much more.

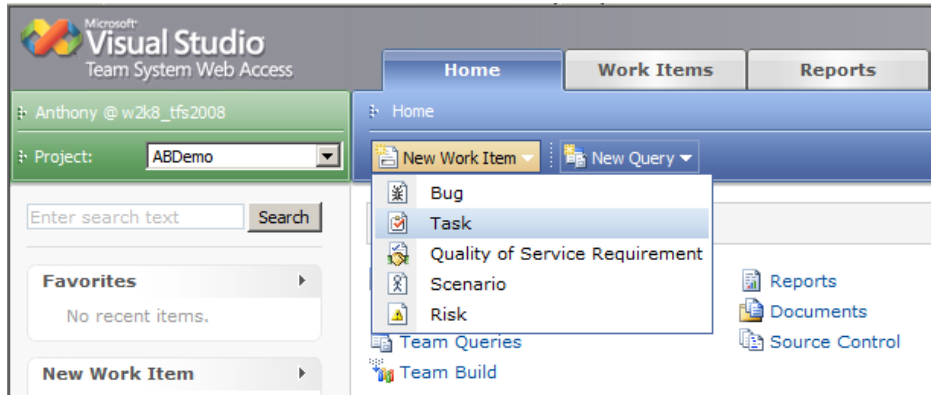


Figure 7: The Team System Web Access interface.

Team System Web Access provides a feature rich user interface for all team members especially those that don't normally work in the Visual Studio IDE. For example, the Team System Web Access interface would be an ideal way to enable your user acceptance engineers to raise change requests or bugs directly into Team Foundation Server. You can download the Studio Team System Web Access 2008 Power Tool from the Microsoft Download Center.

Using the Team Foundation Server API to Create Work Items

Creating work items manually using any of the tools discussed above will suit the majority of members of your development team. The final method for creating work items involves using the Visual Studio 2008 Software Development Kit (SDK) and the Team Foundation Server Object Model to programmatically create work items. This powerful feature enables you to integrate work item tracking with your existing in-house applications.

Often organizations looking to adopt Team System already have a number of in-house developed applications to assist in the software development process. These may include custom bug tracking software or solutions to help manage work. Many of these custom applications are well used and provide great value already. Through the use of the Team Foundation Server API, you can easily modify these custom applications and integrate with Team Foundation Server. For example, your existing bug entry Web application could be easily updated to create a bug work item in Team Foundation Server in preference to storing the information in a custom SQL database.

To illustrate how easy it is to integrate your applications with Team Foundation Server, snippet 1 shows Visual Basic sample code that creates a Task work item.

```
Imports System
Imports System.Collections.Generic
Imports System.Text
Imports System.Collections
Imports Microsoft.TeamFoundation.Client
Imports Microsoft.TeamFoundation.WorkItemTracking.Client

Module CreateWISample

    Sub Main()

        ' Connect to your Team Foundation Server Server
        Dim server As String = "http://w2k8_tfs2008:8080"

        Dim tfs As TeamFoundationServer =
TeamFoundationServerFactory.GetServer(server)

        Dim store As WorkItemStore =
CType(tfs.GetService(GetType(WorkItemStore)), WorkItemStore)

        ' Use the first team project for our example
        Dim workItemTypes As WorkItemTypeCollection =
store.Projects(0).WorkItemTypes

        ' Create a Task work item
        Dim wit As WorkItemType = workItemTypes("Task")

        Dim workItem As New WorkItem(wit)

        ' Set some initial values
        workItem.Title = "This is a new Task Work Item"
        workItem.Description = "This work item has been created using
the Team Foundation Server Object Model"

        ' Save the work item
        workItem.Save()

        Console.WriteLine("Work Item #{0} Successfully Created.",
workItem.Id)

        Console.ReadLine()

    End Sub

End Module
```

Snippet 1: Creating a work item using the Team Foundation Server object model.

Note: The example above is a very minimal example. In a real-world example, you would include error handling and you would also check for invalid fields before saving the work item.

Download the Visual Studio 2008 Software Development Kit (SDK) 1.0 (<http://www.microsoft.com/downloads/details.aspx?familyid=30402623-93ca-479a-867c-04dc45164f5b&displaylang=en>) from the Microsoft Download Center.

DEFAULT WORK ITEMS

When you create a new team project using the project creation wizard, the wizard automatically creates a number of initial work items. These initial work items are defined in the process template you select on the second page of the wizard. Initial work items are significant as they represent the first steps required for each new team project and ensure the required setup and initiation steps are completed as the new project kicks-off.

The MSF for Agile Software Development process template, for example, defines the following 15 initial work items for each new team project. As you read these, it is important to realize these initial work items are not samples, but rather actual tasks you typically need to complete when you start a new project.

MSF for Agile Software Development Default Work Items
Set up: Set Permissions
Set up: Migration of Source Code
Set up: Migration of Work Items
Set up: Set Check-in Policies
Set up: Configure Build
Set up: Send Mail to Users for Installation and Getting Started
Create Vision Statement
Set up: Create Project Description on Team Project Portal
Create Personas
Define Iteration Length
Create Test Approach Worksheet including Test Thresholds
Brainstorm and Prioritize Scenarios List
Brainstorm and Prioritize Quality of Service Requirements List
Set up: Create Project Structure
Create Iteration Plan

Table 3: List of the 15 default work items created with the MSF for Agile Software Development process template.

You can change these default work items to better suit the unique requirements of your team. For example, your organization may not have used a previous work item tracking product, so you may wish to remove the “Migration of Work Items” default work item. Remember a work item cannot be deleted so it would be best to remove this from the process template rather than require that this work item be closed for every new team project. The next section discusses how you can change the default work items to

ensure your next team project creates only work items relevant to your process.

Changing the Default Work Items by Modifying the Process Template

Once you have determined that you wish to modify the initial work items for future team projects, you can modify the process template to create new initial work items as well as remove or modify any of the existing initial work items. The initial work items that are created for each new team project are defined in a file called `workitems.xml`. You can view and modify the contents of this file by exporting one of the process templates from Team Foundation Server using the following steps:

1. Open one of the team editions of Visual Studio 2008 and make sure you are connected to your Team Foundation Server.
2. From the Team menu, select Team Foundation Server Settings and then select Process Template Manager.
3. Select one of the process templates (e.g., MSF for Agile Software Development) and click Download.
4. Save the process template to a convenient location on your local hard disk.

You can find the `workitems.xml` file in `<download location>\MSF for Agile Software Development – v4.2\WorkItem Tracking` folder

Here's an example of the definition of one of the initial work items that is automatically created.

```
<WI type="Task">
  <FIELD refname="System.Title" value="Set up: Set Permissions" />
  <FIELD refname="System.IterationPath"
value="$$PROJECTNAME$$\Iteration 0" />
  <FIELD refname="System.State" value="Active" />
  <FIELD refname="System.Reason" value="New" />
  <FIELD refname="Microsoft.VSTS.Common.Issue" value="No" />
  <FIELD refname="Microsoft.VSTS.Common.ExitCriteria" value="Yes" />
  <FIELD refname="System.Description" value="Add team members to one
of the four security groups: Build Services, Project Administrators,
Contributors, or Readers. To configure security, right-click the team
project in Team Explorer, and select 'Team Project Settings,' 'Group
Membership'" />
</WI>
```

Snippet 2: Sample of a default work item instance defined in the MSF for Agile Software Development process template.

Fortunately Microsoft provides a great utility called the Process Template Editor, which is part of the Visual Studio Team System 2008 Team Foundation Server Power Tools.

You can download it from the Microsoft Download Center at <http://msdn2.microsoft.com/en-us/tfs2008/bb980963.aspx>.

The Process Template Editor provides a convenient graphical interface for editing the default work items and adding new ones:

1. Download and install the Visual Studio Team System 2008 Team Foundation Server Power Tools.
2. Download the ‘MSF for Agile Software Development –v4.2’ process template to your local hard disk following the instructions shown in the previous section.
3. To start the Process Template Editor, from the Tools menu, select Process Editor, select Process Templates, and then select Open Process Template.

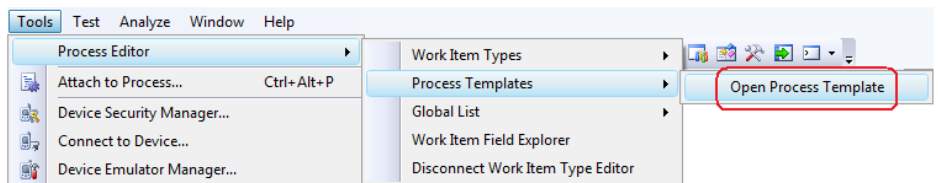


Figure 8: Opening a Process Template using the Process Template Editor.

4. Open the ProcessTemplate.xml file from the folder you downloaded in step 2, e.g., C:\MSF for Agile Software Development - v4.2\ProcessTemplate.xml.
5. Select the “Default Work Items” node in the tree.

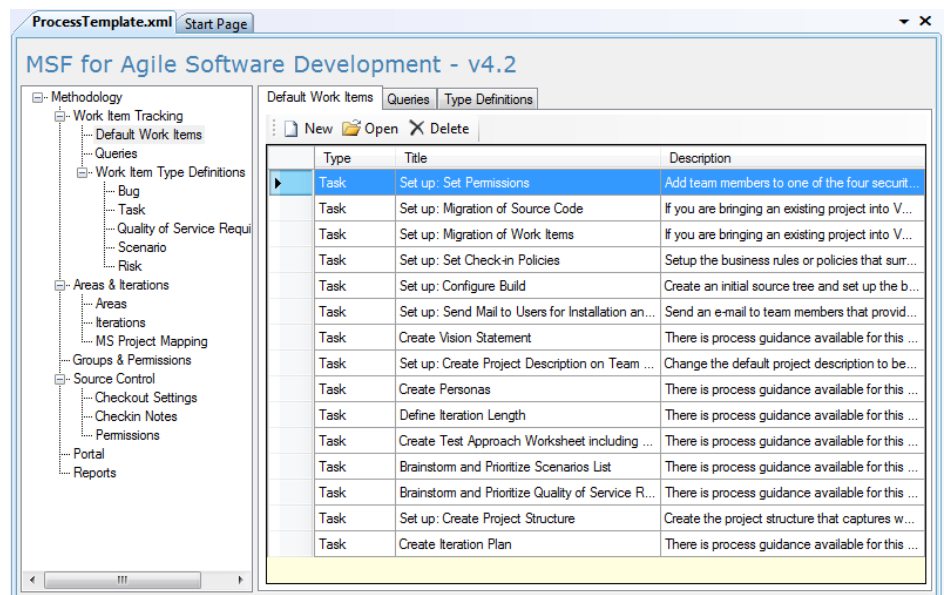


Figure 9: Default work items viewed in the Process Template Editor.

6. You can now create, edit, and remove the default work items using the “New,” “Open,” and “Delete” buttons.

Once you have modified the process template, you should rename the new template so as to not confuse it with the out-of-the-box templates. You will need to upload the modified process template back into Team Foundation Server so you can use your customized process template next time you create a new team project.

To upload your updated process template to Team Foundation Server:

1. From the Team menu, select Team Foundation Server Settings and then select Process Template Manager.
2. Click Upload.
3. Navigate to the folder containing your updated process template and click upload again.

WORK ITEM CUSTOMIZATION

Both the MSF for Agile Software Development and MSF for CMMI Process Improvement process templates have a number of work item types predefined. Table 1, lists the default work item types for both out-of-the-box process templates. Before you look at how to modify work item types, here's an example of why it can be beneficial to create your own work item types.

Consider the graphic artist (GA) on the team who creates the graphics for things like splash screens and toolbar icons. Using a generic Task work item, the majority of the information the GA requires is in a single text box in a field called Description. How often do you think the GA receives all the information and details necessary to complete the task the first time?

Some of the information the GA requires includes color scheme, dimensions, format, purpose, culture, and more. How much time does the GA waste as they seek clarification for each and every assigned task? The solution to this problem is to create a custom work item type to ensure all the necessary information is captured the first time. Consider the time saved when a project manager creates a new "Icon" work item that includes a drop-down list of available file formats and a mandatory field for image dimensions. For an investment of a few hours to create and test a new work item type, the team could realize savings of a few days by reducing the time taken to capture the requirements and time saved not having to redo work due to poor requirements.

One important point to remember is that changes to your process template will only affect new team projects created using your updated process template. Any existing team projects will not be affected.

Working with WITimport and WITexport

During the execution of the Project Creation Wizard, the work item type definitions are read and the types defined for the new team project. Changes made after that to the work item definitions in the process template will not affect any existing team projects. Often you will need to make changes to a work item type after the team project has been created. For example, your project manager wants to add a new field called "TRS Number" to task work items in an existing team project.

You can use two utilities called WITImport and WITExport to import a new work item type and export an existing work item type, respectively. You can use these two tools to make changes to work item types in existing team projects on a project-by-project basis. The following diagram shows how you can use these two utilities. In this example, we are using witexport to export the Task type from an existing team project; we make modifications to the work item type and then import it as a new type using witimport. Of course we could import it as Task and replace the original type, but on this example we want to create a new work item type based on the Task type.

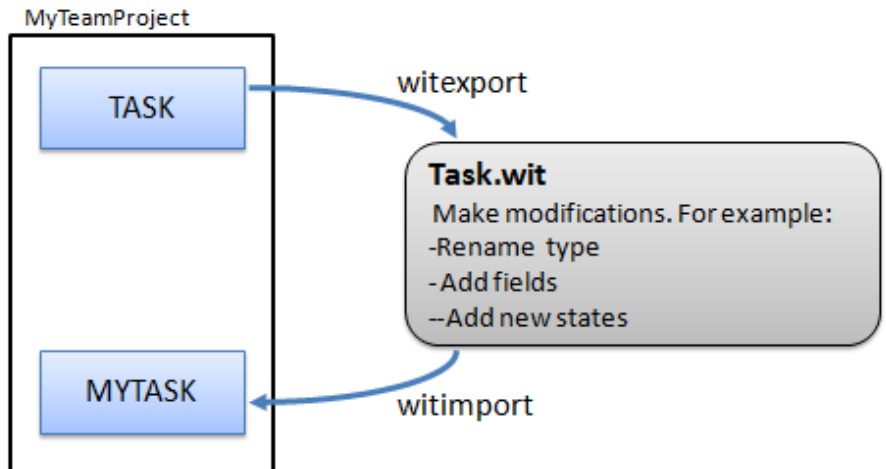


Figure 10: The process of using WitExport and WitImport

You can usually find the utilities, WITImport and WITExport, in C:\Program Files\Visual Studio 9.0\Common7\IDE on any computer that has the Team Explorer installed.

For more information about these two utilities, visit the MSDN Library online:

WITImport — <http://msdn2.microsoft.com/en-us/library/ms253163.aspx>

WITExport — <http://msdn2.microsoft.com/en-us/library/ms253051.aspx>

Defining Work Item Types Using the Process Template Editor

Another capability of the Process Template Editor, discussed in a previous section, is to create and edit work item types. You have a number of options when it comes to editing work item types using the Process Template Editor. You can edit the work item types in a process template, directly on the server, or by editing a work item file. The WITImport and WITExport utilities focus on editing work item templates from an existing team project so this document will examine editing the work item types in the process template.

The following steps show you how to use the Process Template Editor to edit a work item type:

1. Start Visual Studio.
2. From the Tools menu, select Process Editor, Process Templates, and then select Open Process Template.

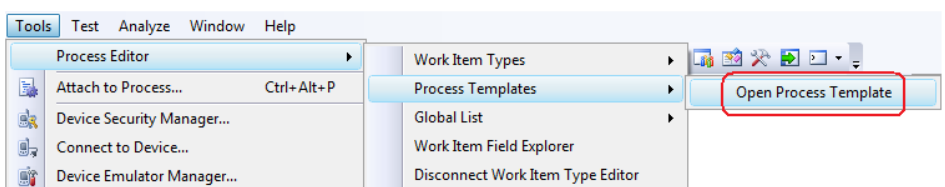


Figure 11: Opening a process template with the Process Template Editor.

3. Open the Process Template.xml file from the folder you previously downloaded, e.g., C:\MSF for Agile Software Development - v4.2\ProcessTemplate.xml.
4. Select “Work Item Type Definitions” in the tree as shown below.

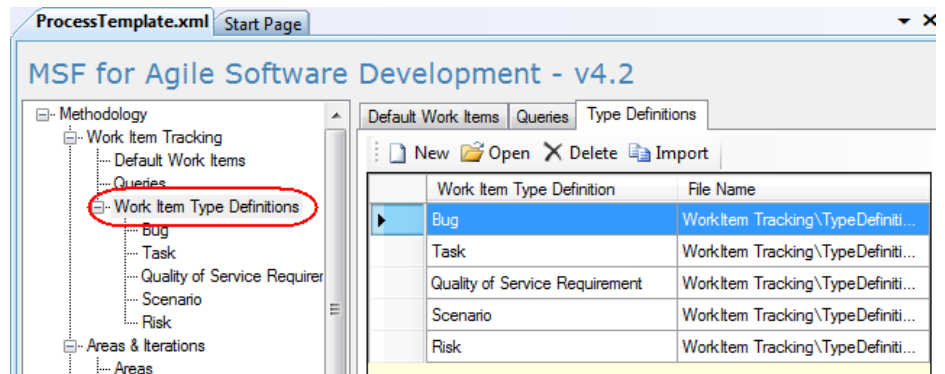


Figure 12: Viewing work item types in the Process Template Editor.

5. Generally when you want to create a new work item, you will want to start with a copy of an existing work item. The Process Template Editor makes this easy by enabling you to choose an existing work item to copy when you click the New button. The New Work Item Type dialog box, shown in Figure 13, illustrates how you can easily copy an existing work item type.

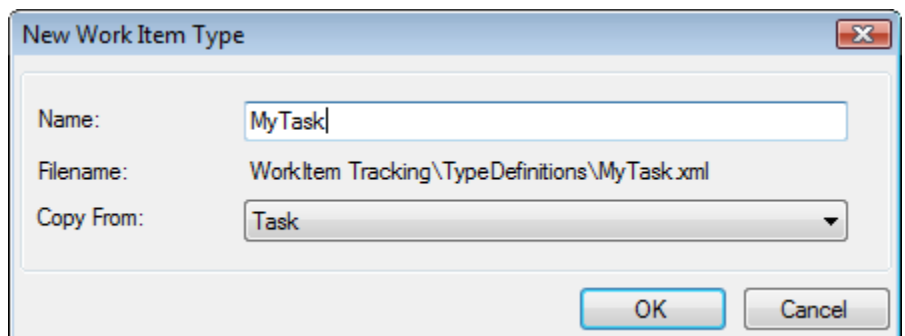


Figure 13: The New Work Item Type dialog box enables you to copy an existing work item type.

6. You can now add and modify fields as well as modify states and transitions using the Process Template Editor.

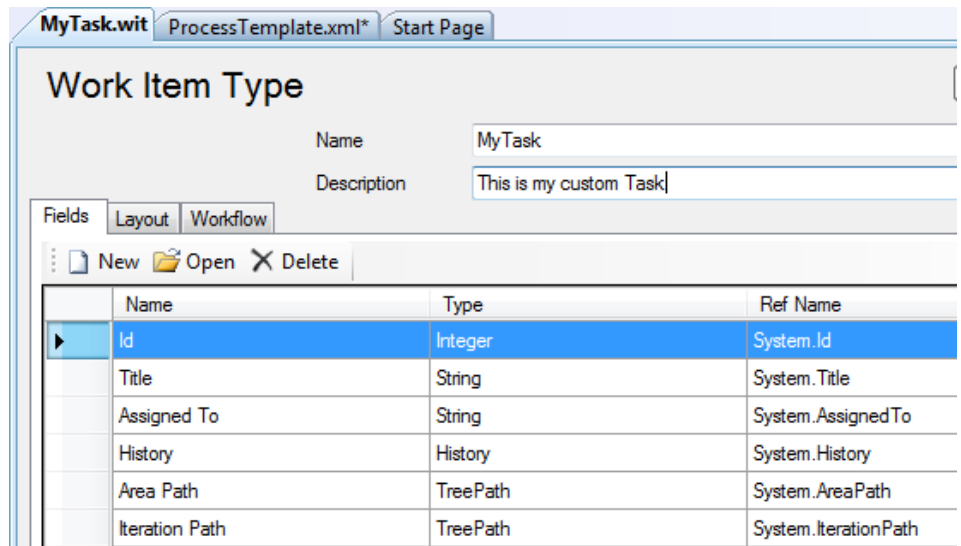


Figure 14: Modifying a work item type in the Process Template Editor.

7. Once you have completed your changes, click Save to first save your work item type, and then save your ProcessTemplate.xml file.

Once you upload the modified process template to your Team Foundation Server, you will be able to create new team projects that will include your modified work item types.

VISIBILITY AND TRACEABILITY

Creating and assigning work items to team members is a great start in managing team workflow. This simple start enables us to define units of work and place them in a queue ready for the team members to read the task and take the necessary action. Once the work items have been created and assigned to our team members, it is important that they are able to quickly and easily find the work items. Project managers also need to be able to determine the status of work items to see what work has been completed, what is outstanding, and if any tasks are currently blocked. This section looks at work item queries and reporting in Team System and how their effective use adds significant value to managing team workflow.

Work Item Queries

There are a number of default work item queries provided with each of the out-of-the-box process queries or you can easily create your own. You can access these templates through the Team Explorer in Visual Studio or through the Team System Web Access Power Tool.

Using Existing Queries

The default process templates include a number of work item queries out of the box including those listed in Table 4:

MSF for Agile Software Development	MSF for CMMI Process Improvement
Active Bugs	Active Bugs
All Issues	All My Team Project Work Items
All Quality of Service Requirements	All Tasks
All Scenarios	Blocked Work Items
All Tasks	Change Requests
All Work Items	Corrective Actions
My Work Items	Customer Requirements
My Work Items for All Team Projects	Development Tasks
Project Checklist	Issues
Resolved Bugs	Mitigation Action Status
Untriaged Bugs	My Work Items
	Product Requirements
	Resolved Bugs
	Reviews
	Risks

Table 4: List of default work item queries.

Creating and Customizing Queries

Team Foundation Server stores all work items in a dedicated database called “TfsWorkItemTracking”. Writing a custom work item query is really just like writing a standard database query with the exception that you use Work Item Query Language (WIQL).

The following walk-through creates a custom work item query to find work items assigned to you that you have closed in the past seven days. You might find this a very useful query to consider adding to your team projects.

While you can easily create the following work item query by modifying the existing My Work Items query, I’ll choose to create this query from scratch.

1. Start Visual Studio.
2. From the Team menu select Add Query.

Visual Studio will create a new query for you with a default clause provided to limit the scope of the query to the current team project.

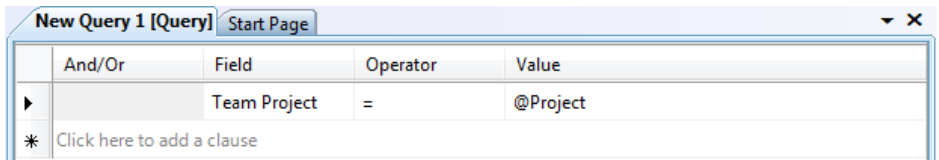


Figure 15: A new work item query includes an initial clause to limit results to the current team project.

3. Click where it says “Click here to add a clause” and select the following values for each cell. (Note: You won’t need to type anything—select values from drop-down lists)
And | Closed Date | >= | @Today - 7

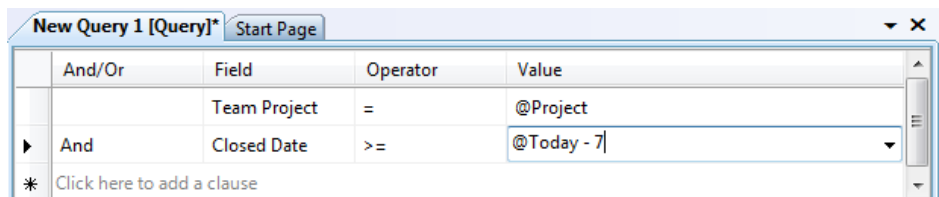


Figure 16: Limiting query results to those closed in the previous seven days.

4. To limit the results to only those work items that have been assigned to you, add another clause with these values.
And | Assigned To | = | @Me

Note: By using “@Me” instead of choosing your name from the drop-down list, other people could use this query.

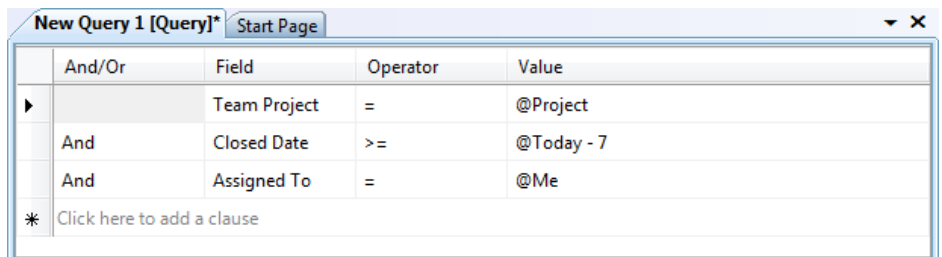


Figure 17: Limiting query results to those assigned to you.

5. If you created this new work item query in a team project where you have recently closed work items, you can test the new query by clicking the “Run Query” button as highlighted in figure 18. You can still run the query if you have not closed any work items in the current team project, but you will receive a “0 results found” message.

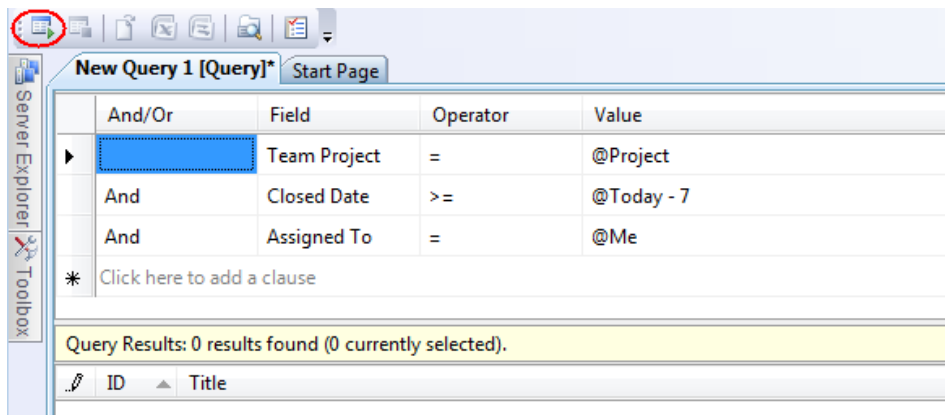


Figure 18: Running your new work item query by clicking the Run button on the toolbar.

6. Now that you’ve completed the new custom query, click the Save button on the toolbar.
7. In the “Save Query As” dialog box, enter “My Work Items—Closed last 7 days” as the name of the query. You then need to choose one of the following locations.
 - Team Query.* Makes your query available to others on your team
 - My Query.* Makes your query available only to you (in My Queries node)
 - File.* Useful for adding the new query to a process template

For this example, I’ll choose “Team Query” so others on the team can use this query.
8. Click “Save” to close the “Save Query As” dialog box.

Your new query should appear in the “Team Queries” node under the team project where you created it. Members of your team can now simply double-click this new query to run it any time they wish.

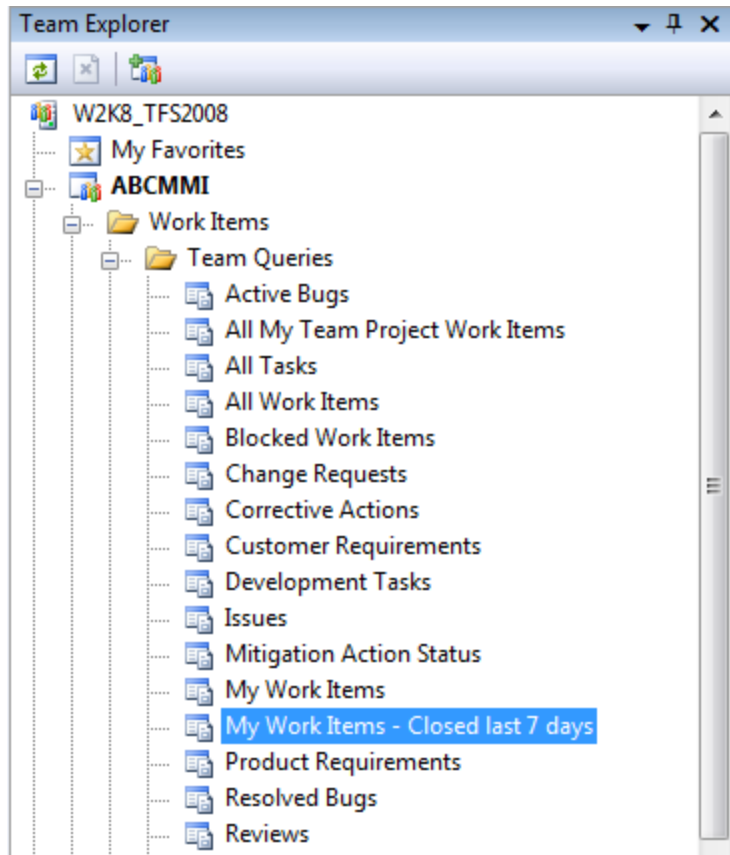


Figure 19: Team queries showing your new work item query.

Creating custom work item queries is a powerful way to enable members of your team to quickly and easily find work items. This is important in ensuring efficient team workflow.

TRACEABILITY

One of the most powerful features of work item tracking is the ability to track all aspects of work being done on the project. Team Foundation Server provides the infrastructure to enable bi-directional traceability, that is, the ability to trace from requirements to end products and from end product back to requirements. This helps you determine that all source requirements have been addressed and that selected work products can be traced to a valid source. To best illustrate the traceability aspects of work item tracking, let's consider the following walk-through.

Defining Scenarios

The Microsoft Solutions Framework (MSF) uses scenarios as the primary form of functional requirements. Typically you will use the "Scenarios.xls" file from the Requirements folder in the Project Portal to create your Scenario work items. You can then further define each scenario work item by using the "Scenario Description" document template, also available in the Requirements folder.

For the purpose of this walk-through, I'll create a simple scenario called "User signs into Web site".

Linking Tasks to Scenarios

Once you have created your scenarios, the next step involves creating tasks to fulfill the scenario. Typically projects have a one-to-many relationship between scenarios and tasks. You'll start by creating a new task called "Create sign-in Web form". One important step you must include is to add a link back to the scenario work item. In figure 20, you can see that the new task work item has been linked back to your scenario (Work item 35 in this example).

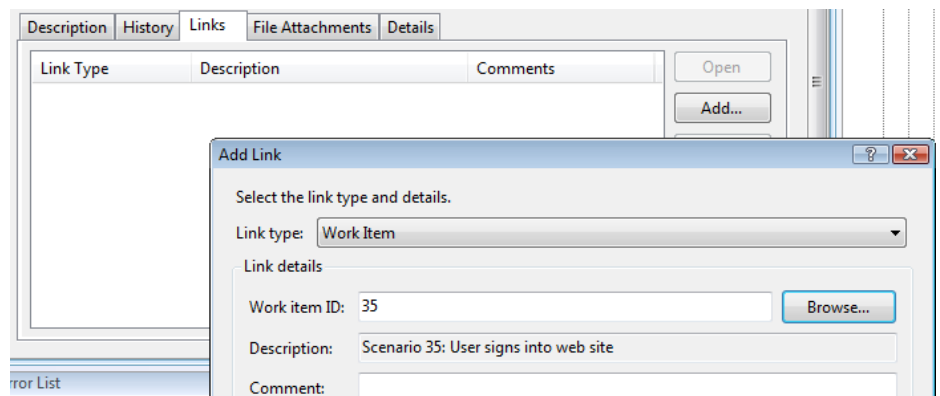


Figure 20: Adding a link to a new work item.

You then create any further tasks required to complete the scenario. Then create tasks for each of the necessary tests and link these back to the various tasks or scenario depending on the scope of the specific test. You should also link back any relevant Quality of Service (QoS) Requirements to the task or scenario.

Linking Changesets to Tasks

When a developer checks their source code into Team Foundation Server, they can link their changeset to the specific work item it relates to. The step of linking the changeset to the work item represents a minor step for the developer checking the code in, but a critical step in terms of ensuring traceability. To help ensure this step is not forgotten, you can use the “Work Item” check-in policy to prompt the developer if they do not link the changeset with a task. Figure 21 shows the Work Items check-in policy applied to the ABDemo team project.

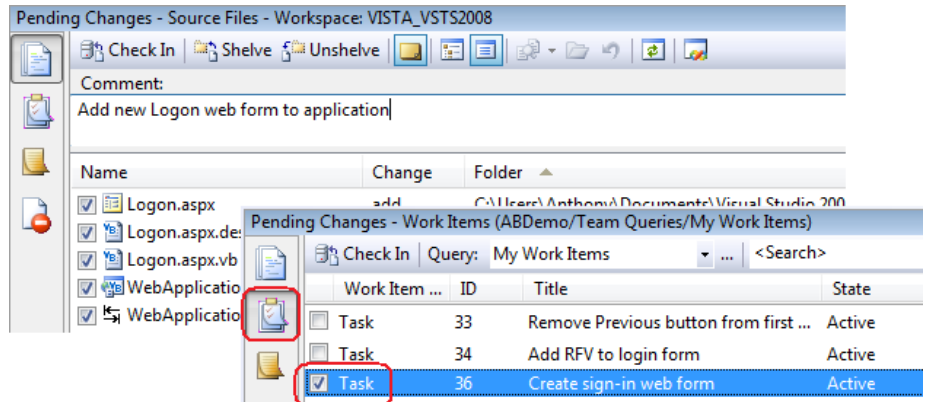


Figure 21: Applying the Work Item check-in policy to a team project

Once the Work Items check-in policy has been configured for a team project, the following warning () is displayed whenever a check-in is attempted that does not reference a work item. This is an invaluable step in ensuring we maintain full traceability through our project.

In Figure 21, you can see that the changeset containing the new logon.aspx Web form is linked to Work Item 36: Create sign-in Web form.

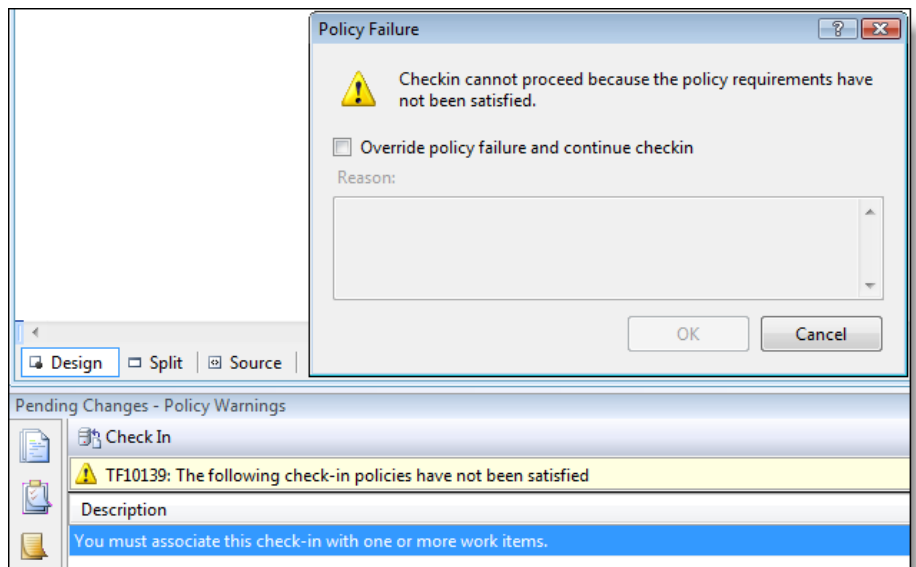


Figure 22: Linking a changeset with a work item during the check-in process.

This document could continue demonstrating examples for work item linking, including work items created from builds and test runs; however, let's examine the benefits of these links in terms of the traceability it provides members of the team.

There are many different ways to illustrate the traceability provided by the simple steps shown above. For example, a developer might look at the version history of the logon.aspx file in the Source Control Explorer and want to know why this work was done. In Figure 23 you can see how linking work items enables the developer to double-click in each of the dialog boxes and trace the specific file version right back to the scenario that required the change. Conversely, you could just as easily go the other way and track the Scenario down to individual changes in a specific version of a file.

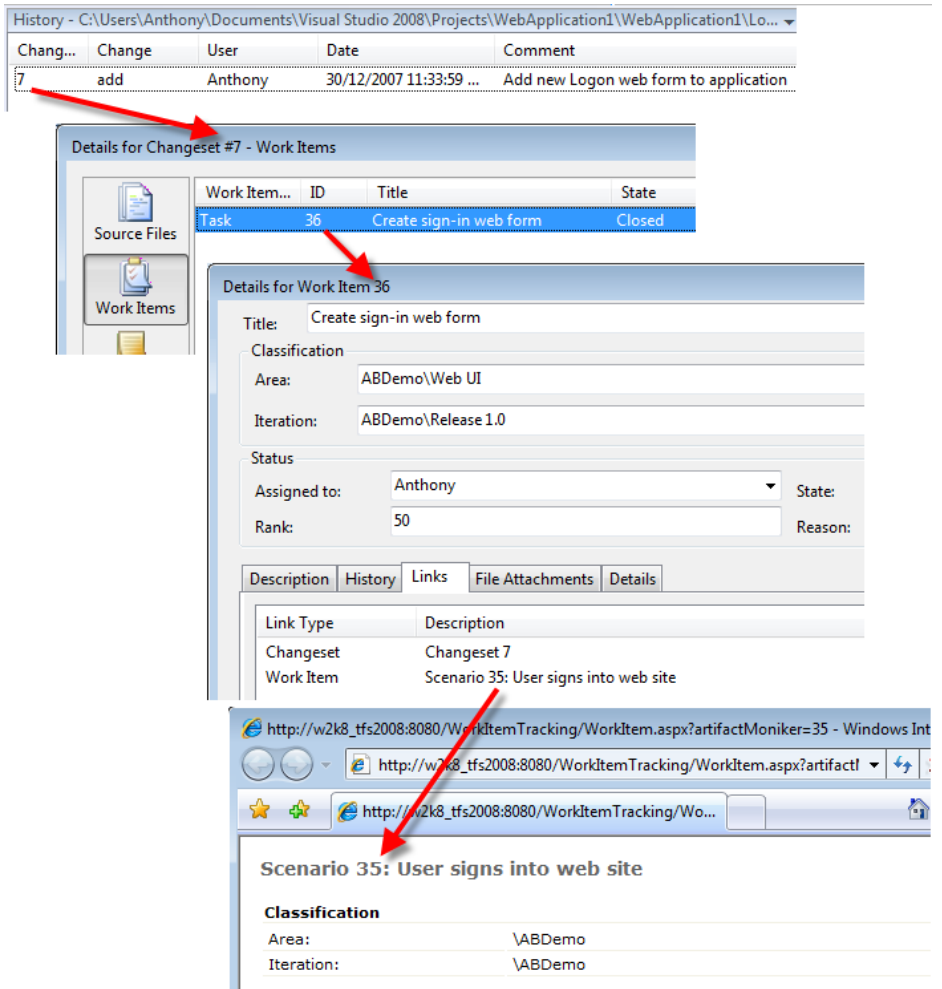


Figure 23: Tracing a file version back to the scenario that required the change.

This example of traceability is just one of many ways team members can view all activities in the development process. In the next section you'll see how you can use the built-in reporting capabilities of Team Foundation Server.

Managing Work across Multiple Team Projects

In some environments, it is possible that you may be regularly working on multiple team projects. Team System makes this convenient for you by providing a "My Favorites" node in Team Explorer.

The My Favorites node enables you to store work item queries from a number of different team projects in a central location. While this may not seem like much, it enables you to quickly and easily access your work item queries without the need to remember which project contains the query you are looking for and places these in a convenient location for quick access.

To add work item queries to the My Favorites node, you can simply copy and paste them from the team project that you have created the queries in.

1. Create a work item query, either from scratch or by modifying one of the existing queries.
2. Save the work item query in one of your team projects.
3. Right-click on the query in the Team Explorer and select Copy from the context menu.
4. Right-click on the My Favorites node near the top of the Team Explorer windows and select Paste from the context menu.

Figure 24 shows an example of a number of custom work item queries being added to the My Favorites node in Team Explorer. The key benefit of this is that you don't need to navigate through individual team projects to run these queries. You can also create folders under the My Favorites node to categorize your favorite queries if you wish.

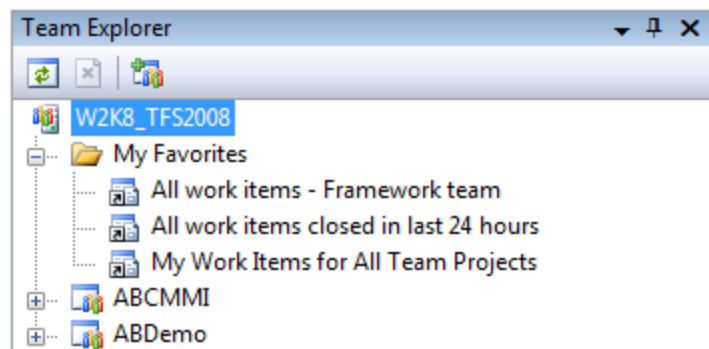


Figure 24: Work item queries added to the My Favorites node in Team Explorer.

Reporting

Reporting is one of the most beneficial features of Team Foundation Server. The reports clearly illustrate the benefits of an integrated application life-cycle product like Team System over a “bundled” group of tools. Reports in Team Foundation Server are generated in near real time from an OLAP cube using SQL Reporting Services.

Work Remaining

The Remaining Work report (figure 25) answers some of the most frequently asked questions in software development... “When will it be ready?”, “How are we doing?” or “Will we ship on time?”

Team Foundation Server reports can be easily viewed through the project portal, the Team Explorer or Team System Web Access. In the example below, once the gradients become apparent, quite early in the iteration, you could reliably extrapolate out the lines to find approximately when all work would be completed. Naturally this assumes a number of things, including steady resource level, a minimum of scope creep, and more, but by the end of the second week you would be able to estimate with a high degree of accuracy when the iteration would be complete.

This report shows a fairly healthy iteration with only eight extra work items being added during the iteration.

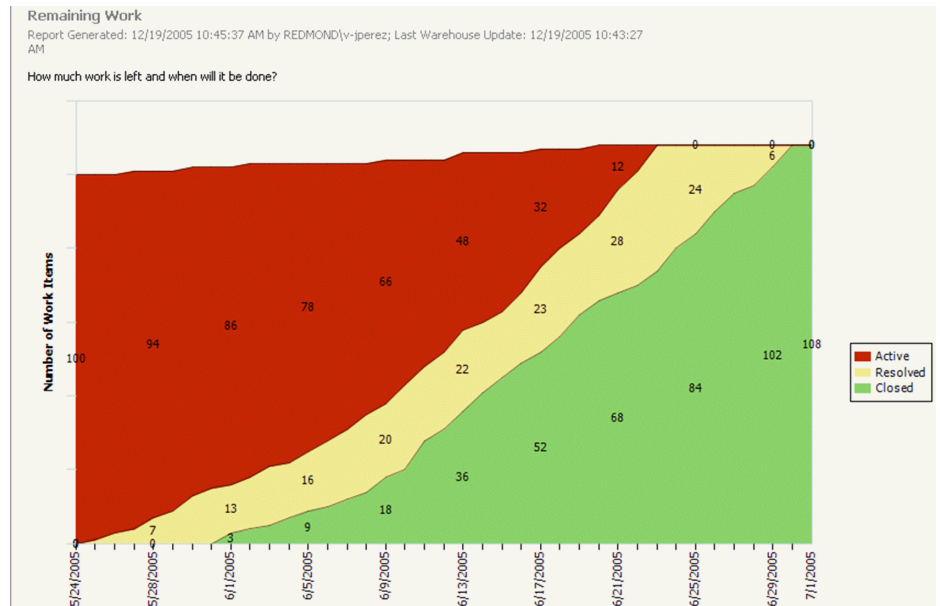


Figure 25: A sample Remaining Work report.

Unplanned Work Report

The Unplanned Work report shows an iteration over a one month period. At the start of the iteration we planned to complete 85 individual work items. Unfortunately during the iteration, an additional 74 work items were added which pushed out much of the planned work. The most common explanations for such a report are:

- Scope creep. Unplanned work is pushing out planned work and work is being added in an ad-hoc fashion. Project managers need to be aware of this as it starts to occur and often deal with the source of the issue early to ensure expected deliverables are met.
- Poor planning. This report could also be the product of a very poor planning process where the actual work needed is not correctly defined at the start of the iteration and is added during the iteration. This is also an issue that a project manager needs to be aware of as soon as possible so corrective action can be taken quickly.

Project Velocity Report

The following Project Velocity report provides a graphic representation of how much work, measuring in closed work items, that the team is doing over a period of time. Let's consider a number of possible reasons for this downturn.

- A competing project has "acquired" a number of your key developers resulting in a sharp decline in the productivity of team. Such a

downturn would represent a significant risk to your project delivery schedule and corrective action would need to be considered quickly.

- The particular project has been completed and most resources have been moved to other projects. Only a few developers remain to complete stabilizing work and some minor bug fixes.

There could be other reasons for the downturn shown in the report but this is sufficient to show the value of the automated reports in managing team workflow. Through the regular examination of the reports available in Team System, project managers are able to effectively manage workflow in a much more responsive manner; ensuring problems are identified and rectified much more quickly.

You find out more about Team Foundation Server Reporting in the MSDN Library (<http://msdn.microsoft.com/library/ms194922>).

CONCLUSION

Microsoft Visual Studio Team System 2008 provides a compelling development environment for all team members involved in the application development life cycle. Through the use of the work item tracking capability of Team Foundation Server, you can define units of work and track the status of this work efficiently and easily. The built-in reporting capabilities of Team Foundation Server, utilizing SQL Reporting Services, provide unprecedented visibility into the activities and progress of your teams than ever before. The out-of-the-box reports greatly assist in managing your team workflow by providing valuable metrics in near real-time that project managers can use to keep the project on track.

As an integrated application life-cycle management solution, Microsoft has delivered the tools needed for achieving unprecedented traceability to development teams. For example, you can track a bug from the build where it is identified, back to the changeset that introduced the code, to the work item that led to the changeset, and then back to the specific scenario the work item was related to. Team System enables you to work effectively as a team and meet the demands of business as your team strives to develop a wide variety of cutting-edge business applications.

With Team System you have the tools and capabilities you need to easily manage your team's workflow from requirements right through to delivery.

ABOUT THE AUTHOR

Anthony Borton is a leading Visual Studio Team System trainer and consultant based in Brisbane, Australia. He is a Team System Microsoft Valued Professional (MVP) and can be reached through his blog at www.myvstsblog.com.

This white paper was developed in partnership with A23 Consulting.