Hands-On Lab

Introduction to Test Case Management with Microsoft Test Manager 2012

Lab version: 11.0.50522.1 (RC)

Last updated: 6/7/2012

* + 1. 

Contents

[Overview 3](#_Toc326729970)

[Exercise 1: Managing Test Suites and Test Cases 4](#_Toc326729971)

[Exercise 2: Analyzing Test Runs 13](#_Toc326729972)

[Exercise 3: Selecting Builds to Test Against 17](#_Toc326729973)

Overview

* 1. Microsoft Test Manager 2012 can be used to organize your test plans, author and manage your test cases, and run manual tests. This lab will provide you with a basic understanding of how Microsoft Test Manager can be used to perform these activities. Also see “*Authoring and Running Manual Tests with Microsoft Test Manager*” if you are interested in these aspects of Microsoft Test Manager.
  2. This hands-on-lab is one out of a number of labs that deal with Microsoft Test Manager 2012. The recommended order to complete the labs in is as follows:
  + Introduction to Test Planning with Microsoft Test Manager 2012
  + Introduction to Test Case Management with Microsoft Test Manager 2012
  + Authoring and Running Manual Tests using Microsoft Test Manager 2012
  + Introduction to Platform Testing with Microsoft Test Manager 2012

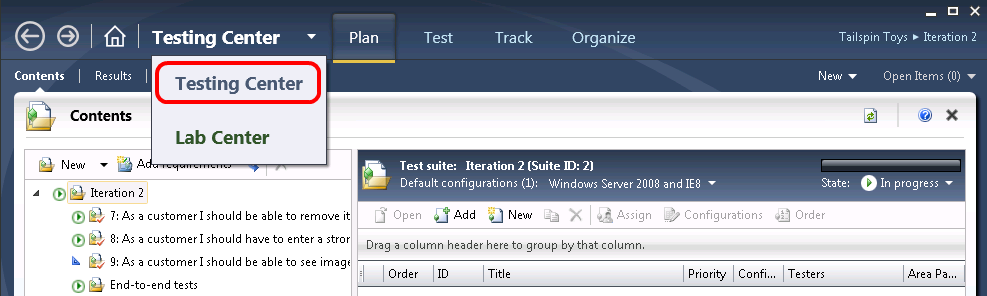
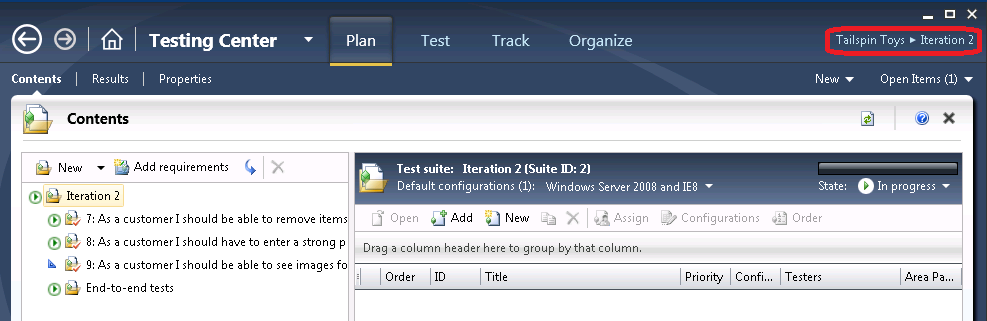
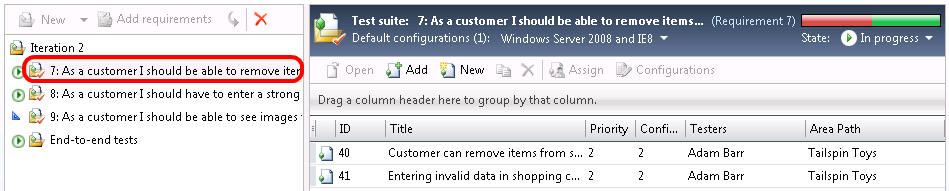
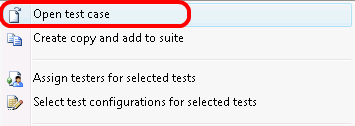
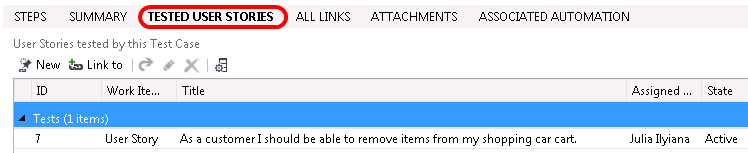
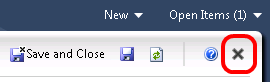
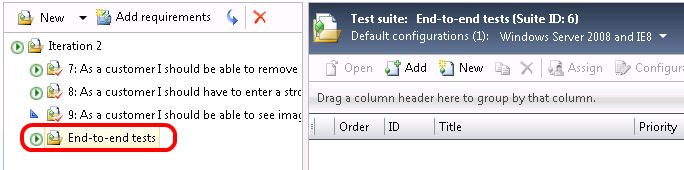
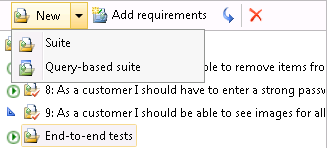
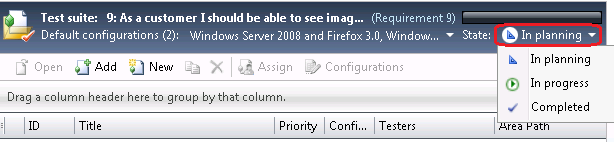
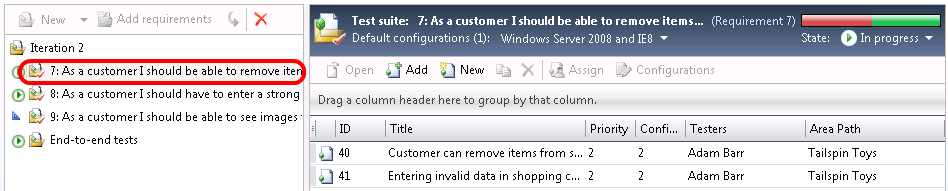
# Prerequisites

In order to complete this lab you will need the Visual Studio 2012 virtual machine provided by Microsoft. For more information on acquiring and using this virtual machine, please see [this blog post](http://aka.ms/VS11ALMVM).

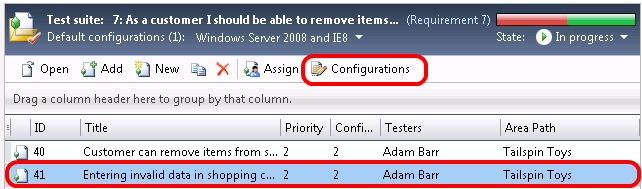
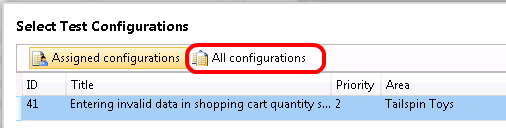
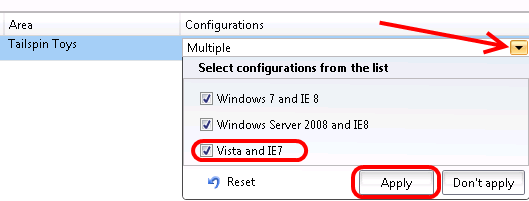
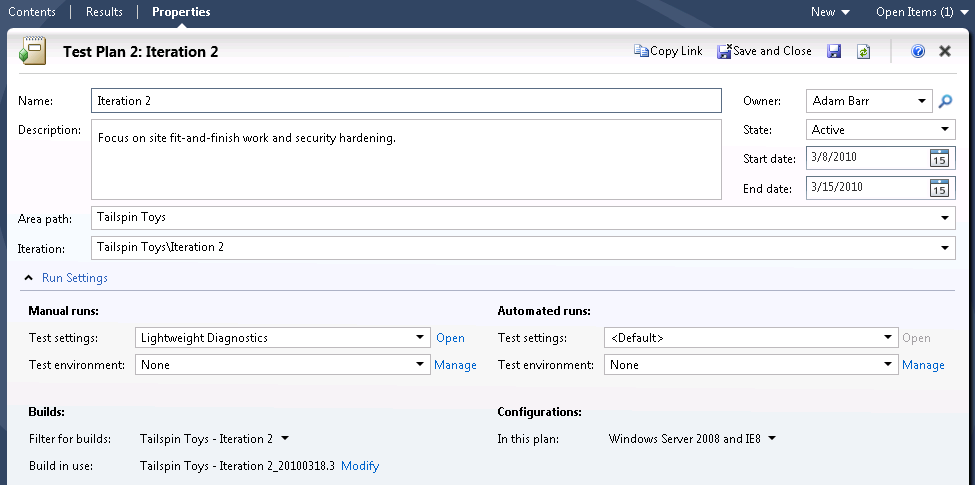
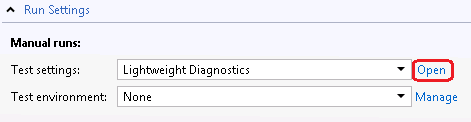
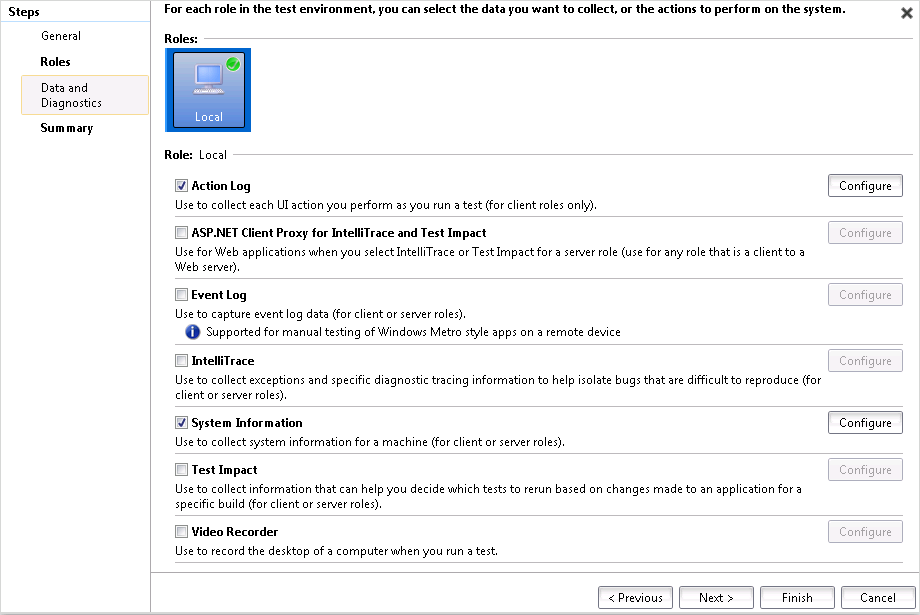
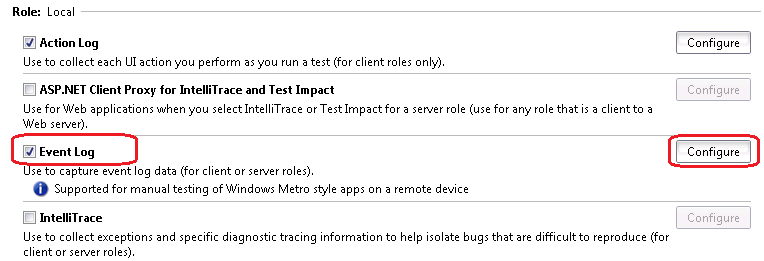
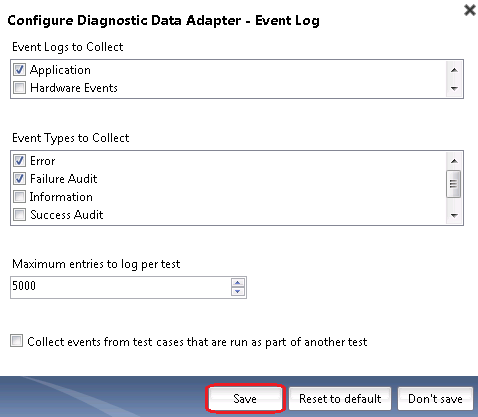
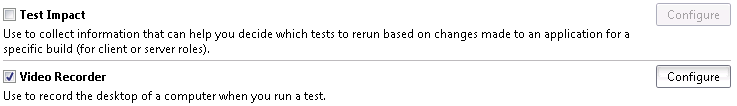
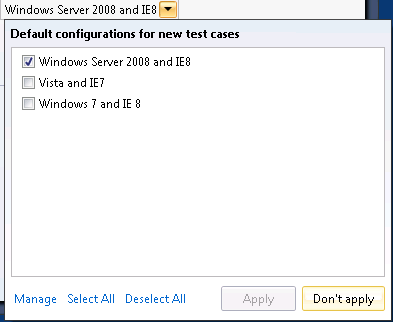
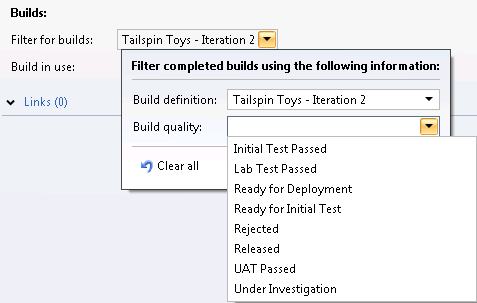
# Exercises

* 1. This hands-on lab includes the following exercises:
  2. Managing Test Suites and Test Cases
  3. Analyzing Test Runs
  4. Selecting Builds to Test Against
  5. Estimated time to complete this lab: **60 minutes**.

Exercise 1: Managing Test Suites and Test Cases

* 1. In this exercise, you will learn how to create and configure a new test plan in Microsoft Test Manager. This test plan can be used, for instance, to test the new release of your software.
  2. Log in as **Julia**. All user passwords are **P2ssw0rd.**
  3. Open Microsoft Test Manager from **Start** | **All Programs** | **Microsoft Visual Studio 2012** | **Microsoft Test Manager**.
     1. **Note:** Microsoft Test Manager allows testers to work with test plans, author and organize manual test cases, execute test cases, file bugs, and post results back to Team Foundation Server.
  4. Select the activity center drop down from the top menu that currently shows Testing Center. The **Testing Center** is used to work with manual test cases and other general test case management. **Lab Center** is used to work with physical or virtual testing labs. For this lab, we will focus on the Testing Center.
     1. 
     2. Figure
     3. Testing Center showing the Plan activity
  5. If you are not in the Test plan view already, click on **Plan** from the main menu. The **Microsoft Test Manager** will connect to the most recently loaded test plan, which in the case of this virtual machine is the test plan named *Iteration 2*.
     1. 
     2. Figure
     3. Test plan contents window for Iteration 2
  6. In the **Contents** panel under the Iteration 2 test suite node, locate the nodes labeled 7, 8, and 9. These nodes are requirements-based **test suites** which contain test cases linked to requirements. Select **test suite 7** to see the list of linked test cases.
     1. 
     2. Figure
     3. Test cases assigned as requirements for a test suite
  7. Open the test case with **ID = 41** by right-clicking on it and selecting **Open Test Case** from the context menu.
     1. 
     2. Figure
     3. Opening a test case
  8. Select the **Tested User Stories** tab and note that this test case links back to a user story.
     1. 
     2. Figure
     3. Tested User Stories tab
  9. Close the test case by selecting the **X** in the top-right corner of the test case window. Be careful not to close the Test Manager application.
     1. 
     2. Figure
     3. Location of Close button (X)
  10. Locate and select the test suite labeled **End-to-End Tests**. This **static test suite** was created to hold arbitrary test cases and other test suites, although it currently empty.
      1. 
      2. Figure
      3. This static test suite is currently empty
  11. The third and final test suite type is the **query-based suite**. Although not represented in this lab, query-based test suites allow work item queries to be constructed in order to gather test cases. For example, imagine a scenario where you want to create a test suite that includes all priority 1 test cases from iteration 1’s test plan.
      1. 
      2. Figure
      3. Query-based suite
  12. Locate and select **test suite 9** and note that its state is set to **In Planning**. The remaining two states are **In Progress** and **Completed**. Only test suites that are set to the In Progress state are shown on the Test activity tab.
      1. 
      2. Figure
      3. Test suite state selection
  13. Locate and select **test suite 7**. Note that you can open, add, create, and remove test cases from the right-hand side of the window.
      1. 
      2. Figure

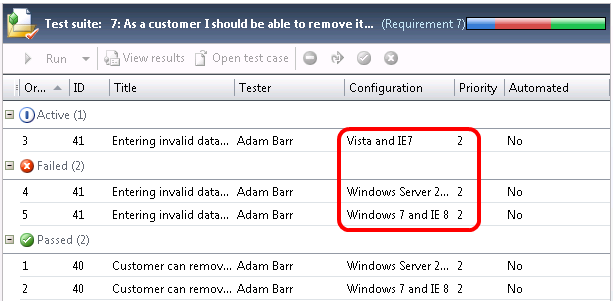
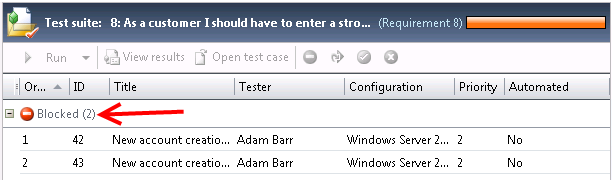
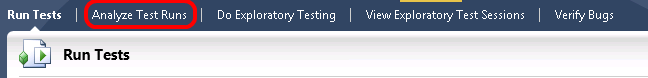
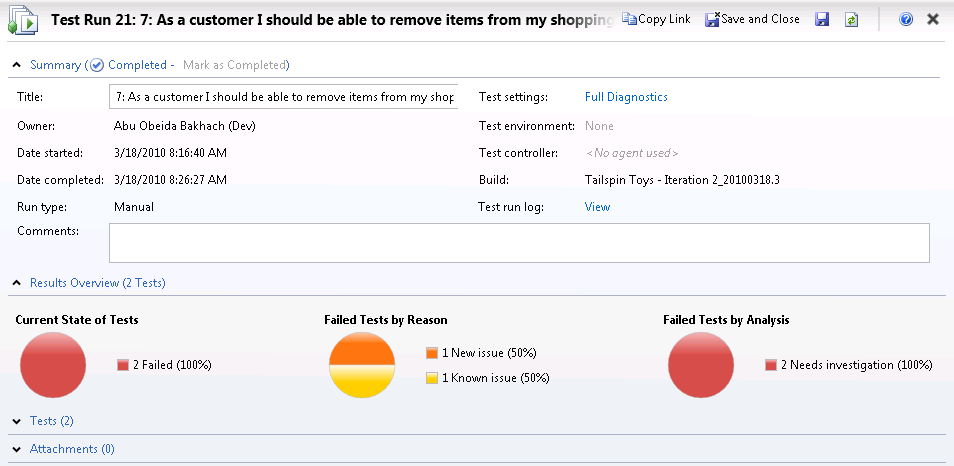
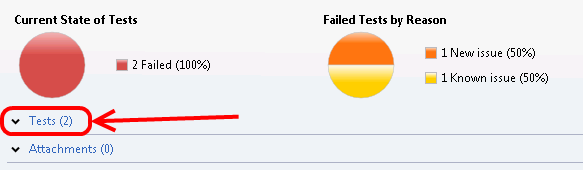
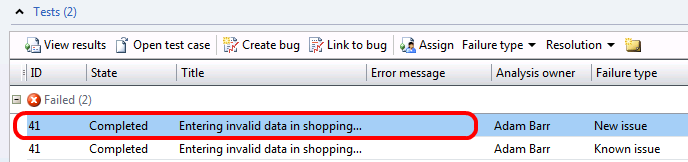
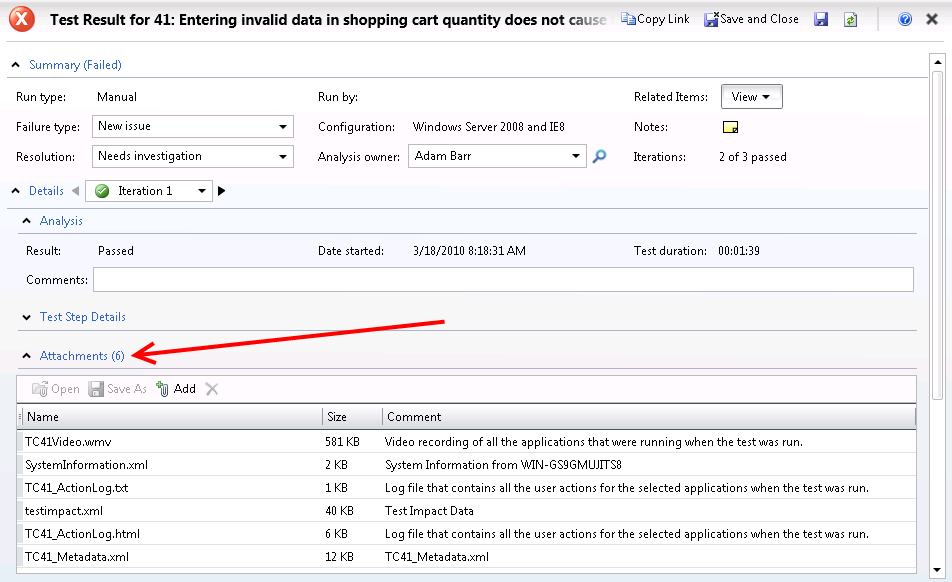
Working with test cases

* + 1. **Note:** To learn more about working with manual test cases, see the lab titled “*Authoring and Running Manual Tests using Microsoft Test Manager 2012*”.
  1. Select the test case with **ID = 41**.
  2. Select the **Configurations** button to load the **Select Test Configurations** window.
     1. 
     2. Figure
     3. Location of Configurations button
  3. Click on the **All Configurations** button.
     1. 
     2. Figure
     3. Location of All Configurations button
  4. Click on **Multiple** under **Configurations** to see all the configuration options available. Select the **Vista and IE7** checkbox and finally select the **Apply** button to continue.
     1. 
     2. Figure
     3. Working with test configurations
  5. Select **Close** button to close the **Select Test Configurations** window.
  6. Select the **Properties** link to load the test plan properties window. The first section of the test plan properties window allows you to edit the basic properties like name, description, area path, iteration, state, and so on. The **Run Settings** section allows you to specify how manual and automated test runs should be setup, what test environment to use, and even the specific build to test against.
     1. 
     2. Figure
     3. Test plan properties window
     4. **Note:** Test settings are particularly important as they specify how and what data will be collected during test runs. If bugs are found during a test run, this data will be used by developers to reproduce and better understand the problem.
  7. Open the test settings for **Manual Runs** by selecting on the **Open** link.
     1. 
     2. Figure
     3. Location of Open link
  8. Select the **Data and Diagnostics** step from the left-hand side of the test settings window. This allows you to select which **Data Diagnostic Adapters** you want to utilize. For example, the Video Recorder will record the screen as seen by the tester during test runs.
     1. 
     2. Figure
     3. Data and Diagnostics options
  9. Enable the **Event Log** diagnostic adapter and then select the **Configure** button.
     1. 
     2. Figure
     3. Event Log diagnostic adapter
  10. In the **Configure Diagnostic Data Adapter – Event Log** window, note that you can specify which event logs and event types to collect from. Select the **Save** button to continue.
      1. 
      2. Figure
      3. Configuring the Even Log diagnostic adapter
  11. Enable the **Video Recorder** diagnostic adapter.
      1. 
      2. Figure
      3. Video Recorder diagnostic adapter
  12. Select the **Finish** button to return to the test plan properties window.
  13. **Test environments** are also managed from the test plan properties window. Test environments are either physical or virtual environments used to run tests against or to collect data. For example, when testing a web application, it may be necessary to collect data from both the client and the server at the same time as components of that system reside in different environments. For this test plan, there is just one local test environment.
  14. **Test configurations** describe which platforms to test against during test runs. Select the drop down that currently has Windows Server 2008 and IE8 selected to see the options available. Select the **Don’t Apply** button when finished.
      1. 
      2. Figure
      3. Specifying testing platforms
  15. The **Builds** section of the test plan properties window selects the build that test results and bugs will be filed against. Select the drop down labeled **Filter For Builds**. In this case there is only one build definition, but you can see that filtering can be done against both definition and quality.
      1. 
      2. Figure

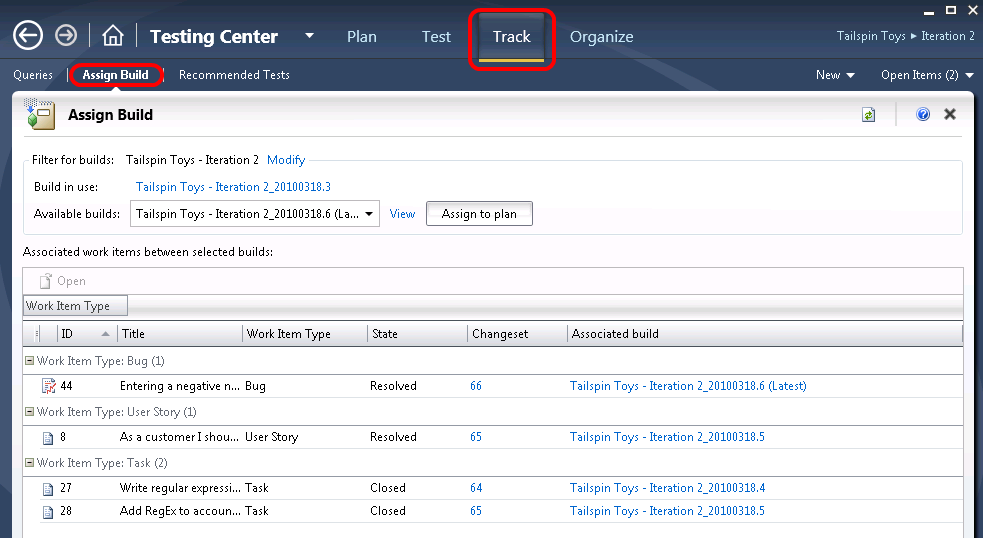
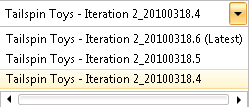
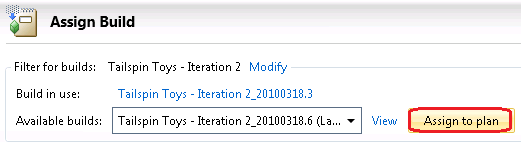
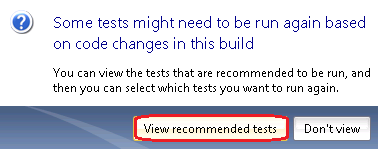
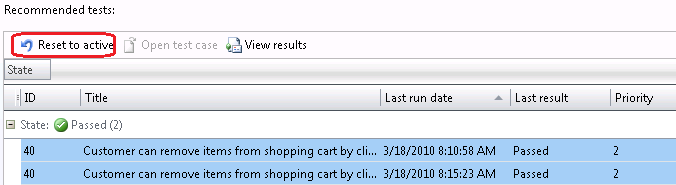
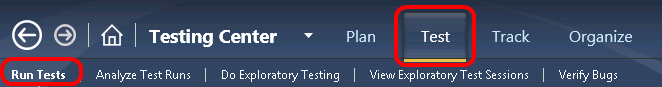
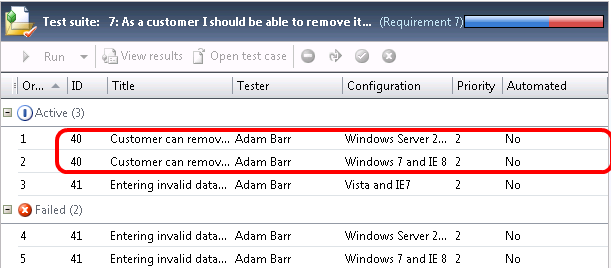
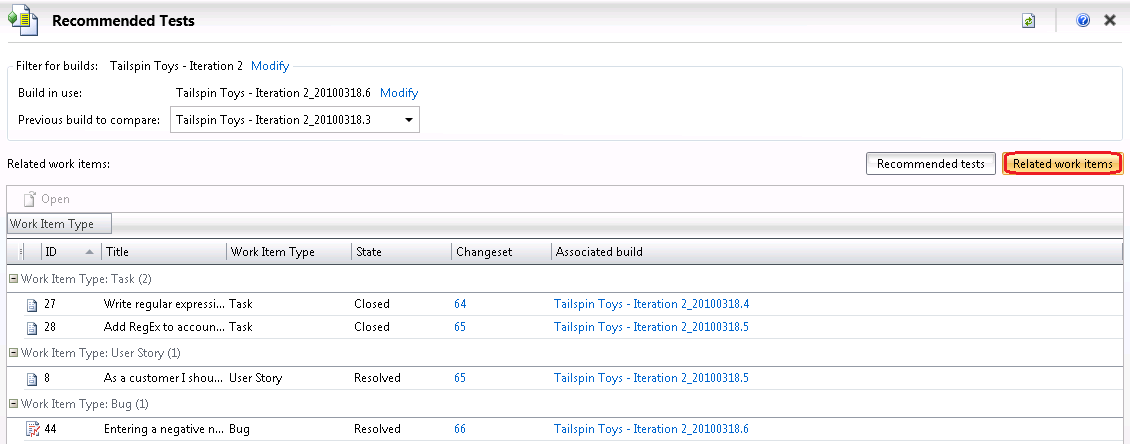
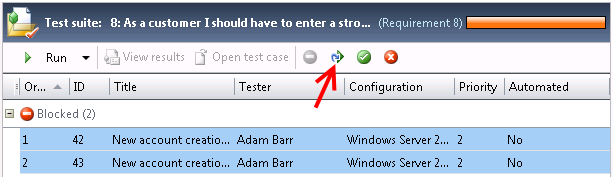
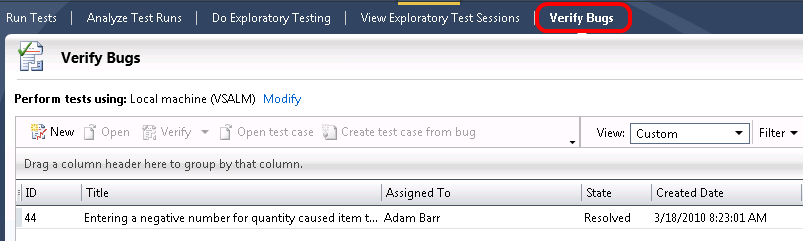
Filtering for builds and build quality

* 1. Press the **Escape** key to cancel the selection and return to the test plan properties window.
  2. Finally, note that a specific build is setup for test runs.
     1. 
     2. Figure
     3. Build version specified for testing
  3. Close the test plan properties window by selecting the **X** in the top-right corner of the test plan properties window. Be careful not to close the Test Manager application.

Exercise 2: Analyzing Test Runs

* 1. In this exercise, you will learn how to use the Test activity to analyze test runs.
  2. In Microsoft Test Manager, select the **Test** tab to open the test activity used by testers. By default, the **Run Tests** window is shown.
  3. Select the **test suite 7** node to view the test runs and their states. Note that the test case with ID = 41 is represented three times, one for each configuration.
     1. 
     2. Figure
     3. Viewing test runs for a test suite
  4. Select the **test suite 8** node to view its test runs. Note that two of the test cases are listed as being blocked. **Blocking** a test case is an action that can be performed by the tester to indicate that they are unable to perform the test case, perhaps due to something such as a user story not being implemented yet or a dependency requirement not working. For example, if the test case is to create an account on a web application, and some other related component prevents the tester from even attempting the action, the tester can indicate that they are blocked.
     1. 
     2. Figure
     3. Blocked test cases
  5. Select the **Analyze Test Runs** link to view all of the completed test runs.
     1. 
     2. Figure
     3. Location of Analyze Test Runs link
  6. Open the test run with **ID = 21** by double-clicking on it. A summary shows start and completion timestamps, type, build version, and so on.
     1. **Note:** If no test runs are listed, you may need to adjust the View filter to include manual runs or modify the date range to include all test runs.
     2. 
     3. Figure
     4. Results from a test run
  7. Expand the **Tests** area to expose the results of the test run.
     1. 
     2. Figure
     3. Expanding the Tests area
  8. Double-click on the first test result to view the details.
     1. 
     2. Figure
     3. Viewing test run results
  9. In the test results view under **Details | Attachments**, note that there are a number of attachments that were collected during the test run, such as system information and a video.
     1. 
     2. Figure
     3. Test results example
  10. Close the test results window by selecting the **X** in the top-right corner of the test results window. Be careful not to close the Test Manager application.

Exercise 3: Selecting Builds to Test Against

* 1. In this exercise, you will learn how to use the Track activity to navigate and select the builds to test against.
  2. In Microsoft Test Manager, select the **Track** tab to open the track activity. By default, the **Queries** window is loaded.
  3. Select the **Assign Build** link to view the available builds and the impacted work items between the selected build and the current build.
     1. 
     2. Figure
     3. Assigning builds for testing
  4. In the **Available Builds** drop down, select the last build in the list (**Tailspin Toys – Iteration 2\_20100318.4**).
     1. 
     2. Figure
     3. Selecting a build
  5. Look at the associated work items to see what work was done during the selected build. Note that a single work item to “Write regular expression for strong password” was closed.
     1. 
     2. Figure
     3. Associated work items
  6. Select the **Tailspin Toys – Iteration 2\_20100318.5** build from the **Available Builds** drop down. There are three associated work items for this build, one of which is the same work item as before.
  7. Select the **Toys – Iteration 2\_20100318.6** build from the **Available Builds** drop down and select the **Assign To Plan** button. This will help provide an indication to the testers that this build warrants doing a full test run. Future test runs will use the new build by default.
     1. 
     2. Figure
     3. Assigning a new build to the test plan
  8. When notified that some tests may need to be run again based on code changes, choose to **view the recommended tests**.
     1. 
     2. Figure
     3. Option to view recommended tests
  9. These recommendations are made possible by **Test Impact Analysis**, which is able to determine when code changes impact previously executed tests. For example, a test that was successfully executed against an earlier build may be recommended to be executed again if it is determined that (a), code has changed and (b), that it is in the code path exercised by that test.
     1. **Note:** Test Impact Analysis can be configured to run in the background when the application being tested is written in managed code (.NET Framework 2.0 or higher).
  10. Select all recommended tests that are impacted by this latest build and select the **Reset To Active** button. This resets the state of test cases within the test plan to the Active state.
      1. 
      2. Figure
      3. Resetting test case state to active for recommended tests
  11. Return to the **Test** activity window and select the **Run Tests** link.
      1. 
      2. Figure
      3. Return to the Test activity
  12. Select **test suite 7**. Note that the two test cases there were previously in the Passing state are now Active again.
      1. 
      2. Figure
      3. Test cases were rest to Active state
  13. Return to the **Track** activity window, select the **Recommended Tests** link, and select the **Related Work Items** button. Note that a number of work items have been closed or resolved since build **Tailspin Toys – Iteration 2\_20100318.3**, including a user story work item.
      1. 
      2. Figure
      3. Related work items
  14. Return to the **Test** activity window and select **test suite 8**, which is associated with the impacted user story that we just looked at in the previous step.
  15. Select the blocked test cases and then select the **Reset** **to active** button to unblock them. Their state goes from Blocked to Active.
      1. 
      2. Figure
      3. Location of Reset button
  16. Select the **Verify Bugs** link. Although we will not do so in this lab, this is where testers could select previously reported bugs and re-run the tests. If the tests do not pass, the testers would re-assign the bugs back to the developers. You may have to change to **view** to **Custom** to see the bugs.
      1. 
      2. Figure
      3. Verifying bugs

To give feedback please write to [VSKitFdbk@Microsoft.com](mailto:VSKitFdbk@Microsoft.com)

Copyright © 2012 by Microsoft Corporation. All rights reserved.