opentext™

OpenText™ Application Quality Management

Software version: 25.1

Tutorial

Go to Help Center online

https://admhelp.microfocus.com/alm/

Document release date: January 2025

Send Us Feedback

Let us know how we can improve your experience with the Tutorial.
Send your email to: admdocteam@opentext.com

Legal Notices

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Disclaimer
Certain versions of software accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. This software was acquired on September 1, 2017 by Micro Focus and is now offered by OpenText, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

Contents

Tutorial

The tutorial is a self-paced guide that instructs you how to organize and manage all phases of the application lifecycle. To successfully complete this tutorial, you should perform the tutorial in the order in which the information is presented.

Tutorial Quick Reference

This tutorial contains the following lessons:

"Specify releases and cycles" on page 15	Learn to define releases and cycles and monitor their progress and quality.
"Specifying Requirements" on page 19	Learn to define requirements, view the requirements tree, and convert requirements to tests.
"Planning Tests" on page 30	Learn to create a test plan tree, define test steps, define test configurations, link test configurations to requirements, and automate manual tests.
"Running Tests" on page 53	Learn to define test sets, schedule test runs, and run manual and automated tests.
"Adding and Tracking Defects" on page 97	Learn to add new defects, update defects, and manage defects.
"Alerting on Changes" on page 113	Learn to track changes made to your requirements, tests, and defects as you perform your project testing.
"Analyzing Data" on page 119	Learn to monitor the application lifecycle management process by creating reports and graphs.

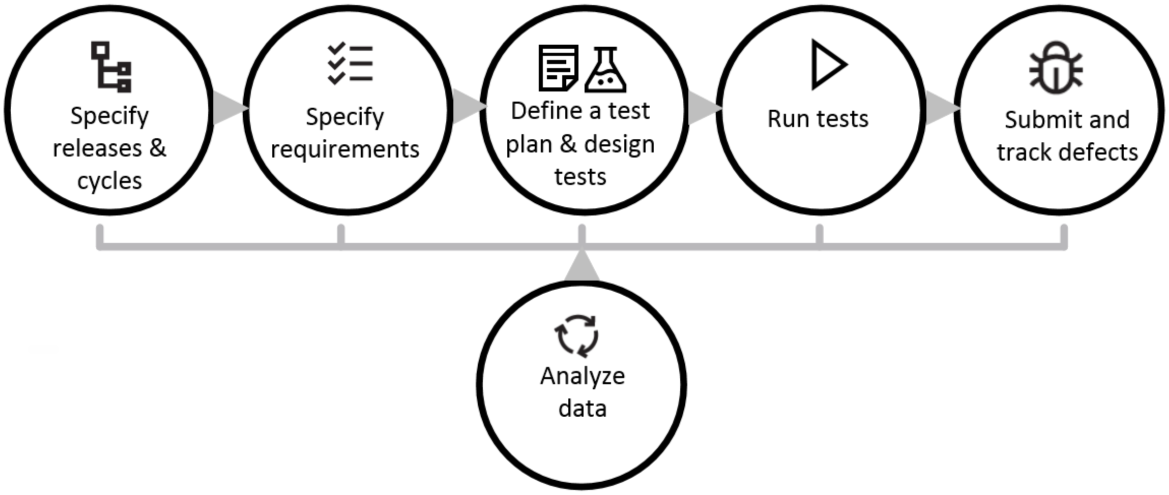
"Creating Libraries and Baselines" on page 147	Learn to create libraries and baselines, and how to compare baselines to track changes in your project.
"Customizing Projects" on page 156	Learn to set up project users, and how to create project fields and lists.

Introducing

OpenText Application Quality Management helps you organize and manage all phases of the application lifecycle management process, including defining releases, specifying requirements, planning tests, executing tests, and tracking defects.

The Process

The application lifecycle management process with OpenText Application Quality Management includes the following phases:



Phase	Description
Specify releases and cycles	Develop a release-cycle management plan to help you manage application releases and cycles efficiently.
Specify requirements	Define requirements to meet your business and testing needs.
Design a test plan and design tests	Based on the project requirements, you can build test plans and design tests.
Run tests	Create a subset of the tests in your project designed to achieve specific test goals. Execute scheduled tests to diagnose and resolve problems.
Submit and track defects	Submit defects and track their repair progress.
Analyze data	Throughout the process, you can generate reports and graphs to assist you in “go/no-go” decisions about your application readiness.

Start

Start OpenText Application Quality Management from your Web browser using the URL.

To start OpenText Application Quality Management:

1. Verify tutorial prerequisites.

   Before you begin the lessons in this tutorial, verify that you have the appropriate prerequisites. For details, see "Tutorial " on page 6.

2. Open your Web browser and type your ALM URL to open the Application Lifecycle Management Options window.

   http://<ALM server name>[<:port number>]/qcbin. Contact your system administrator if you do not have the correct path.

3. For Single-Sign-On users:

   a. If the user discovery page is displayed, add your user name or email address as specified in ALM. Click **Submit**.

   b. In the IDP page, add your IDP credentials. Click the log in button.

4. In the Application Lifecycle Management Options window, click the **ALM Desktop Client** link.

   Each time OpenText Application Quality Management is run, it checks the version. If it detects a newer version, it downloads the necessary files to your machine.

5. In the ALM Login window, type a user name and authenticate.

   In the **Login Name** box, type **alex_alm**.

   Skip the **Password** box. A password was not assigned for the user.

   Click the **Authenticate** button. OpenText Application Quality Management verifies your user name and password and determines which domains and projects you can access.

> **Note:** If ALM was configured for external authentication, the Name and Password fields do not appear in this window. Continue with step 6.

6. In the **Domain** list, select **DEFAULT**. In the **Project** list, select **ALM_Demo**. If more than one **ALM_Demo** project is listed, contact your ALM site administrator to determine which project to use. Click the **Login** button.

Next steps:

- "Specify releases and cycles" on page 15

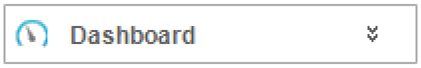
Window

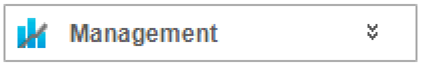
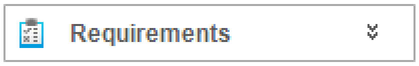
In this exercise, you will explore the OpenText Application Quality Management modules and their common elements. You will also learn how to navigate the online help.

To explore the OpenText Application Quality Management window:

1. Explore the modules.

   Click the following sidebar buttons:

Button	**Description**
 Dashboard	Includes the following modules: • **Analysis View**. Enables you to create graphs and reports. • **Dashboard View**. Enables you to create dashboard pages, in which you can view multiple graphs in a single display.

Button	Description
 Management	Includes the following modules: • **Releases**. Enables you to define releases and cycles for the application management process. • **Libraries**. Enables you to define libraries to track changes in your project, reuse entities in a project, or share entities across multiple projects.
 Requirements	Includes the following modules: • **Requirements**. Enables you to manage requirements in a hierarchical tree-structure. Requirements can be linked to other requirements, tests, or defects. • **Business Models**. Enables you to import business process models, and test the quality of the models and their components. Access to this module is dependent on your OpenText Application Quality Management license.

Button	Description
	Includes the following modules: • **Test Resources**. Enables you to manage test resources in a hierarchical tree-structure. Test resources can be associated with tests. • **Business Components**. Depending on your license, you may also have access to the Business Components module. This module enables subject matter experts to drive the quality optimization process using Business Process Testing, the OpenText test automation solution. • **Test Plan**. Enables you to develop and manage tests in a hierarchical tree-structure. Tests can be linked to requirements and defects. • **Test Lab**. Enables you to manage and run tests. After running tests, you can analyze the results. • **Test Runs**. Enables you to view the results of executed tests.
	Enables you to add defects, determine repair priorities, repair open defects, and analyze the data.

2. Explore the common elements.

   All the OpenText Application Quality Management modules have common elements. For example, click the **Defects** sidebar button.

   Each of the ALM modules contains the following key elements:

   - **ALM common toolbar**. This toolbar is accessible from all modules and contains the following buttons:

Button	Description
‹ ›	Navigates to your previous/next view in ALM.
⚙ ⌄	Provides commands that you can run from each of the ALM modules.
?	Enables you to open the online help and additional online resources. It also enables you to display version information for each ALM client component.
<domain, project, user>	Details of the current domain, project, and user.
⏻	Logs you out of your current project and returns you to the Application Lifecycle Management Login window.

- **Module menu bar**. Displays the menus from which you select commands in the current ALM module.
- **Module toolbar**. This is located below the menu bar. It contains buttons for frequently used commands in the current ALM module.

3. View help topics.

   a. To view the help topic of the Defects module window, click the **Defects** sidebar button. The Defects module is displayed. Click ?. The help topic opens in a separate window.

   b. To view the Application Lifecycle Management Help Center home page, click the masthead or click **Help Center Home** under **Explore** in the footer. The Help Center home page opens.

   The Help consists of guides and references, available online, in PDF format, or both.

   c. Select **Get Started > Content & PDFs** to display the available resources.

   d. Click the **Close** (×) button.

Specify releases and cycles

You begin the application lifecycle management process by specifying releases and cycles. A release represents a group of changes in one or more applications that will be available for distribution at the same time. Each release can contain a number of cycles. A cycle represents a development and QA cycle based on the project timeline. The releases and cycles have defined start and end dates.

You can organize and track your upcoming releases by defining a hierarchical release tree containing releases and cycles. In this lesson, you will add a release to an existing release tree, and then add cycles to the release.

Define releases and cycles

In this exercise, you will define a release and then add cycles to the release. Releases and cycles each have start dates and end dates. The date range for a cycle must be contained within the date range of the release.

To define a release and its cycles:

1. Open the ALM_Demo project.

   If the **ALM_Demo** project is not already open, log in to the project. For details, see "Start" on page 10.

2. Display the Releases module.

   On the ALM sidebar, under Management, select **Releases**.

3. Create a new release folder.

   a. In the releases tree, select the root **Releases** folder. Click the **New Release Folder** button. The New Release Folder dialog box opens.

   b. In the **Release Folder Name** box, type Service Packs.

   c. Click **OK**. The **Service Packs** release folder is added to the releases tree.

   d. In the **Description** box in the right pane, type the following description for the release folder: This folder contains service pack releases.

4. Add a release.

   a. In the releases tree, make sure that the new **Service Packs** release folder is selected.

   b. Click the **New Release** button. The New Release dialog box opens.

   c. In the **Name** box, type Service Pack 1.

   d. In the **Start Date** box, click the down arrow and select yesterday's date. In the **End Date** box, click the down arrow and select the date two months from today's date.

   e. In the **Description** box, type the following description for the release: This release is the first service pack release.

   f. Click **OK**. The **Service Pack 1** release is added to the **Service Packs** release folder.

5. Add a cycle to the release.

   a. In the releases tree, make sure that the **Service Pack 1** release is selected.

   b. Click the **New Cycle** button. The New Cycle dialog box opens.

   c. In the **Name** box, type Cycle 1 - New Features.

   d. In the **Start Date** box, click the down arrow and select yesterday's date. In the **End Date** box, click the down arrow and select the date a month from today's date.

   e. In the **Description** box, type the following description for the cycle: This cycle tests new features added for this service pack.

   f. Click **OK**. The **Cycle 1 - New Features** cycle is added to the **Service Pack 1** release.

6. Add a second cycle to the release.

   a. In the releases tree, right-click the **Service Pack 1** release and choose **New Cycle**. The New Cycle dialog box opens.

b. In the **Name** box, type Cycle 2 - Full.
c. In the **Start Date** box, click the down arrow and select the date one month and a day from today's date. In the **End Date** box, click the down arrow and select the date two month from today's date.
d. In the **Description** box, type the following description for the cycle: This cycle fully tests all application features.
e. Click **OK**. The **Cycle 2 - Full** cycle is added to the **Service Pack 1** release.

View releases and cycles

You can view the status of your releases and cycles. ALM shows a high-level overview of the progress of the currently selected release or cycle. It also shows the number of defects opened over the course of the currently selected release or cycle. You can also view the number of outstanding defects.

In this exercise you will learn how to display the progress and graphs of a selected release and cycle.

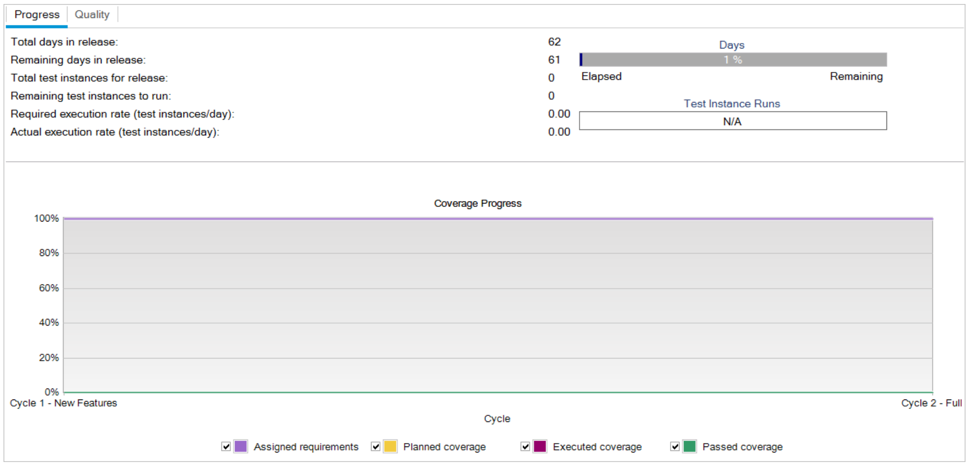
To view releases and cycles:

1. Make sure the **Releases** module is displayed.

   On the ALM sidebar, under **Management**, select **Releases**.

2. Display the Progress graph for the Service Pack 1 release.

   In the releases tree, select the **Service Pack 1** release, located in the **Service Packs** release folder. In the right pane, click the **Status** tab. The Progress tab is displayed by default.



Total days in release:	62
Remaining days in release:	61
Total test instances for release:	0
Remaining test instances to run:	0
Required execution rate (test instances/day):	0.00
Actual execution rate (test instances/day):	0.00

The Progress tab displays the progress of the release based on requirement coverage, elapsed and remaining time, and actual and remaining test instances to run. As you have not yet created requirements or tests, the information in the Coverage Progress graph indicates 0% progress.

3. Display the Progress graph for a cycle.

   In the releases tree, select the **Cycle 1 - New Features** cycle, located in the **Service Pack 1** release.

   In the right pane, click the **Progress** tab. You can see that the information available is similar to that available for the release, but at the cycle level. As in the case of the release, you have not yet created requirements and tests, therefore the information in the Coverage Progress graph indicates 0% progress.

Specifying Requirements

Requirements describe in detail what needs to be solved or achieved to meet the objectives of your application under development.

You define the requirements in ALM by creating a requirements tree in the Requirements module. This is a hierarchically graphical representation of your requirements. You can group and sort requirements in the tree, monitor the progress in meeting requirements, and generate detailed reports and graphs.

In this lesson, you will create requirements in an existing requirements tree. You will then assign the requirements to a cycle in the releases tree. You will also learn how to convert requirements to tests.

In this lesson, you will learn about:

Defining Requirements

In this exercise, you will define requirements for testing the functionality of reserving cruises in Mercury Tours.

To define a requirement:

1. Open the ALM_Demo project.

   If the **ALM_Demo** project is not already open, log in to the project. For details, see "Start" on page 10.

2. Display the Requirements module.

   a. On the ALM sidebar, under **Requirements**, select **Requirements**.

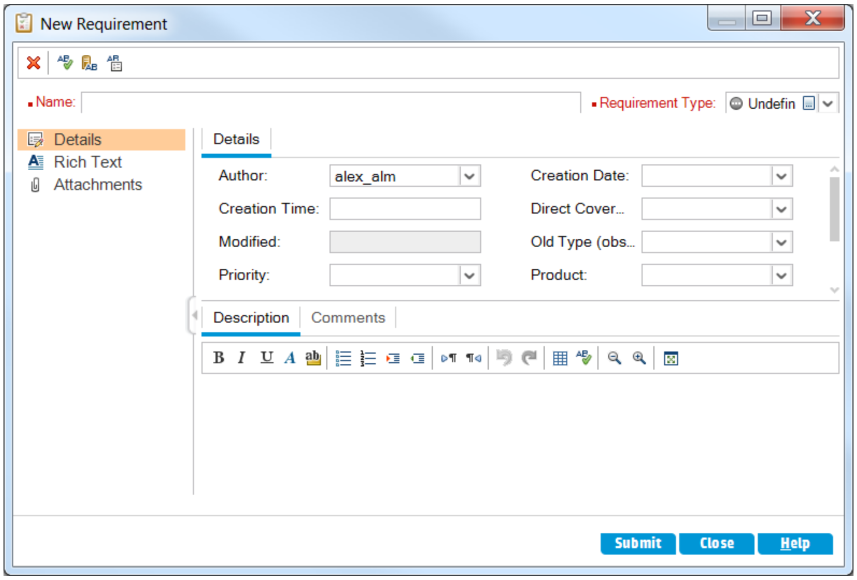
   b. Choose **View > Requirements Tree** to display requirements in a tree.

3. Select the Mercury Tours Application requirement.

   Expand the **Requirements** root requirement and select the **Mercury Tours Application** requirement.

4. Create a new requirement.

   a. Click the **New Requirement** button. The New Requirement dialog box opens.



Requirement Type: Undefin

Author: alex_alm

- b. In the **Name** box, type **Cruise Reservation**.
- c. In the **Requirement Type** box, select **Functional**. Each requirement belongs to a requirement type. The requirement type to which a requirement belongs determines which fields are available for the requirement. Your project administrator can modify existing types and add new types.
- d. In the Details tab, type or select the following:

  Priority: 4-Very High

  Product: Mercury Tours Web Site
- e. Click **Submit**.
- f. Click **Close** to close the New Requirement dialog box. The **Cruise Reservation** requirement is added to the requirements tree under the **Mercury Tours Application** requirement.

5. Add the Cruise Search child requirement.
   - a. In the requirements tree, make sure that the new **Cruise Reservation** requirement is selected.
   - b. Click the **New Requirement** button to add a requirement below **Cruise Reservation**. The New Requirement dialog box opens.
   - c. In the **Name** box, type Cruise Search.
   - d. In the **Requirement Type** box, select **Functional**.
   - e. In the Details tab, type or select the following:

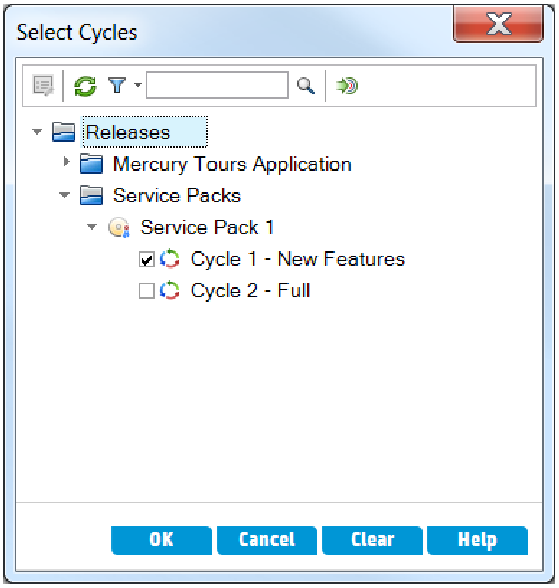
     Priority: 4-Very High

     Product: Mercury Tours Web Site
   - f. Click **Submit**.
   - g. Click **Close** to close the New Requirement dialog box. The **Cruise Search** requirement is added as a child of the **Cruise Reservation** requirement.

6. Add the Cruise Booking child requirement.
   a. In the requirements tree, make sure that the **Cruise Reservation** requirement is selected.
   b. Click the **New Requirement** button to add a requirement below **Cruise Reservation**. The New Requirement dialog box opens.
   c. In the **Name** box, type Cruise Booking.
   d. In the **Requirement Type** box, select **Functional**.
   e. In the Details tab, type or select the following:

      Priority: 4-Very High

      Product: Mercury Tours Web Site
   f. Click **Submit**.
   g. Click **Close** to close the New Requirement dialog box. The **Cruise Booking** requirement is added as a child of the **Cruise Reservation** requirement.
7. Assign the requirements to a cycle.
   a. In the requirements tree, select **Cruise Reservation**.
   b. Choose **Requirements > Assign to Cycle**. The Select Cycles dialog box opens.



Select Cycles

Releases
- Mercury Tours Application
- Service Packs
  - Service Pack 1
    - ☑ Cycle 1 - New Features
    - ☐ Cycle 2 - Full

c. Locate the **Service Packs** releases folder. Under **Service Pack**, select the check box for the **Cycle 1 - New Features** cycle.

d. Click **OK** to close the releases tree.

e. Click **Yes** to assign the requirement and its sub-requirements to the cycle.

8. Assign additional requirements to a cycle.

   a. In the requirements tree, under **Mercury Tours Application**, select **Online Travel Booking Services**.

   b. Choose **Requirements > Assign to Cycle**. In the Select Cycles dialog box, locate the **Service Packs** releases folder. Under **Service Pack 1**, select the check box for the **Cycle 1 - New Features** cycle. Click **OK**.

   c. **Version Control**: If the Check Out dialog box opens, click **OK**.

   d. Click **Yes** to assign the requirement and its sub-requirements to the cycle.

Viewing Requirements

You can change the way requirements are displayed. In this exercise, you will learn how to zoom in and out of the requirements tree, rearrange the requirement order, and display and filter requirements in the requirements grid.

To view requirements:

1. Make sure the **Requirements** module is displayed.

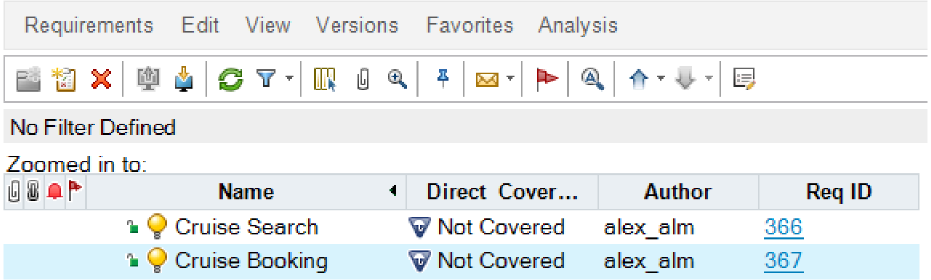
   a. If the Requirements module is not displayed, on the ALM sidebar, under **Requirements**, select **Requirements**.

   b. Choose **View > Requirements Tree** to display requirements in a tree..

2. Zoom in and out of the requirements tree.

a. Select **Cruise Reservation** in the requirements tree.

b. To zoom in, click **Zoom** and choose **Zoom In**. The requirements tree displays only the child requirements of **Cruise Reservation**.



No Filter Defined

Zoomed in to:

Name	Direct Cover...	Author	Req ID
Cruise Search	Not Covered	alex_alm	366
Cruise Booking	Not Covered	alex_alm	367

c. To reverse the zoom-in action and display the entire requirements tree, click **Zoom** and choose **Zoom Out To Root**.

3. Rearrange the order of requirements in the requirements tree.

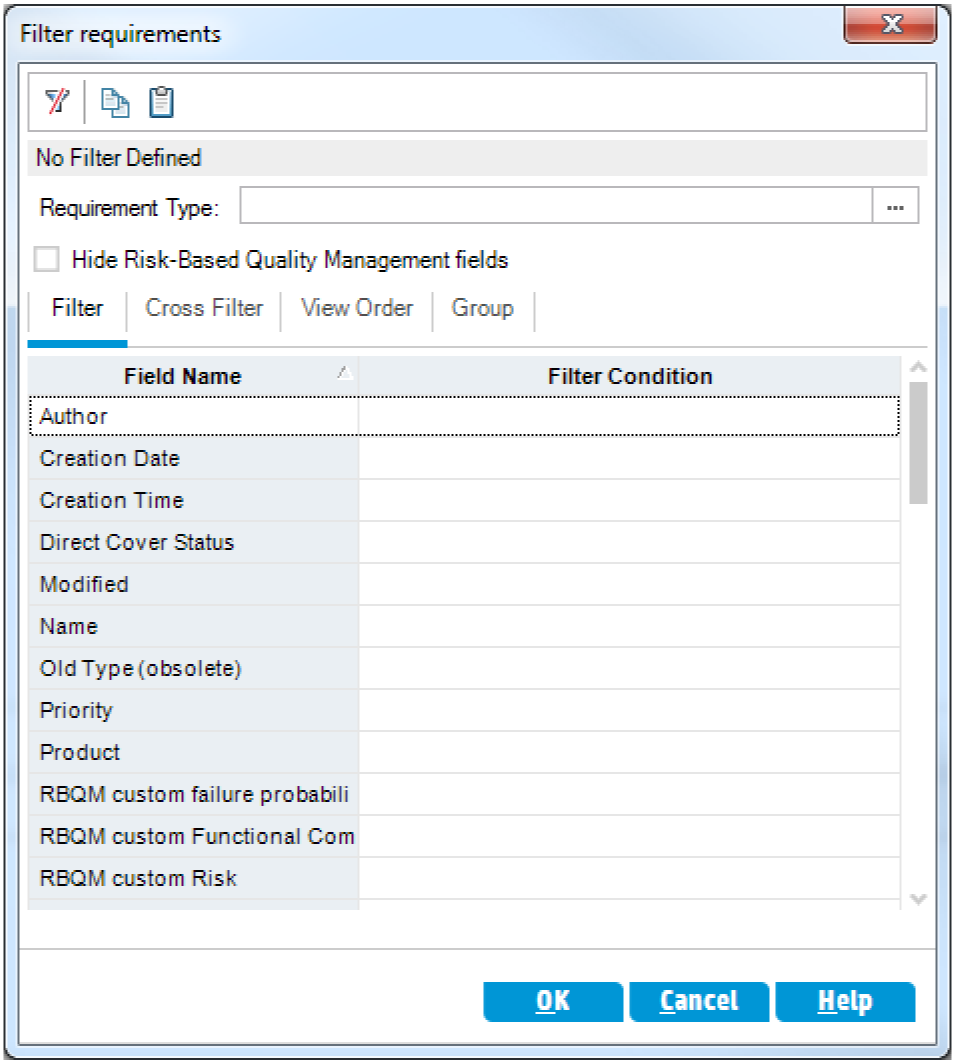
   ALM adds requirements to the requirements tree in order of creation. To rearrange the order, select the **Cruise Search** requirement and click the **Move Down** button. The Cruise Search requirement moves down below the Cruise Booking requirement.

4. View requirements in the requirements grid.

   Choose **View > Requirements Grid** to display requirements in a flat nonhierarchical view. Each line in the grid displays a separate requirement.

5. Define a filter to view requirements created on a specific date.

   a. Click the **Filter** button. The Filter dialog box opens.



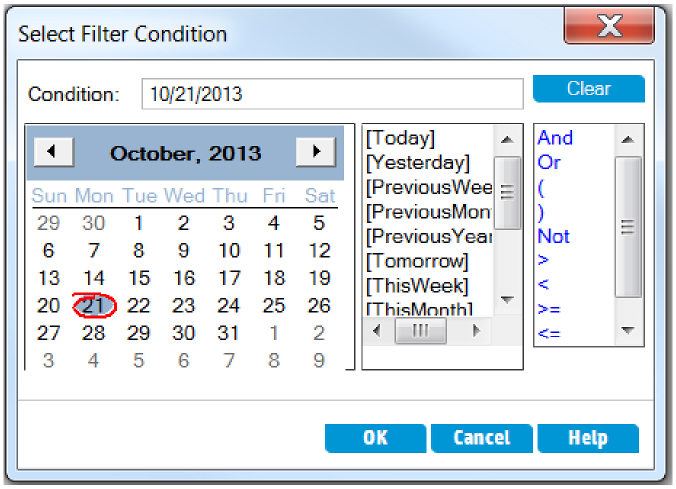
No Filter Defined

Requirement Type:

□ Hide Risk-Based Quality Management fields

Field Name	Filter Condition
Author	
Creation Date	
Creation Time	
Direct Cover Status	
Modified	
Name	
Old Type (obsolete)	
Priority	
Product	
RBQM custom failure probabili	
RBQM custom Functional Com	
RBQM custom Risk	

   b. For the **Creation Date** field, click the **Filter Condition** box. Click the down arrow button. The Select Filter Condition dialog box opens, displaying today’s date in the calendar.



Condition: 10/21/2013

October, 2013

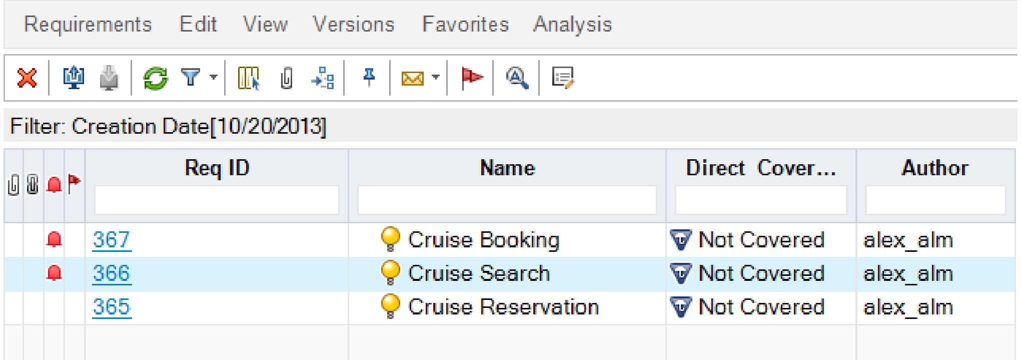
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

c. Select the date on which you added requirements.

d. Click **OK** to close the Select Filter Condition dialog box.

e. Click **OK** to apply your chosen filter.

f. The Requirements Grid displays the requirements you added.



Filter: Creation Date[10/20/2013]

Req ID	Name	Direct Cover...	Author
367	Cruise Booking	Not Covered	alex_alm
366	Cruise Search	Not Covered	alex_alm
365	Cruise Reservation	Not Covered	alex_alm

Converting Requirements to Tests

After you create the requirements tree, you can use the requirements as a basis for defining your test plan tree in the Test Plan module.

You can use the Convert to Tests wizard to assist you when designing your test plan tree. The wizard enables you to convert selected requirements or all requirements in the requirements tree to subjects or tests in the test plan tree.

In this exercise, you will convert the **Cruise Reservation** requirement to a subject in the test plan tree, and the child requirements of **Cruise Reservation** to tests in the Cruise Reservation subject folder.

To convert a requirement to a test:

1. Make sure the **Requirements** module is displayed.

   If the Requirements module is not displayed, on the ALM sidebar, under **Requirements**, select **Requirements**.

2. Select a requirement.

   a. Choose **View > Requirements Tree** to display requirements in a tree.

   b. In the requirements tree, select **Cruise Reservation**.

3. Open the Convert to Tests wizard.

   Choose **Requirements > Convert to Tests**. The Step 1 dialog box opens.

4. Choose an automatic conversion method.

   Select the second option, **Convert lowest child requirements to tests**, to convert the selected requirement to a subject folder, and its subrequirements to tests.

5. Start the conversion process.

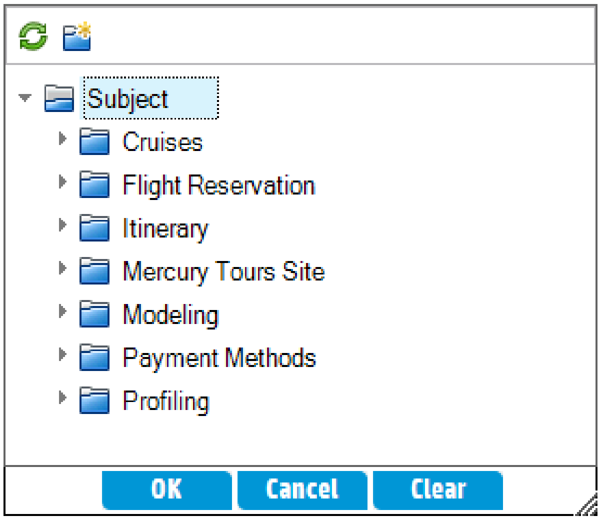
   a. Click **Next** to begin converting the requirements. When the conversion process is complete, the results are displayed in the Step 2 dialog box.

   b. Click **Next**. The Step 3 dialog box opens.

6. Choose the destination subject path.

   a. In the **Subject** box, click the down arrow button. A dialog box displaying the test plan tree opens.

   b. In the test plan tree, select the **Cruises** subject.



c. Click **OK** to close the Select Destination Subject dialog box. The **Subject** box now indicates this test plan.

7. Finalize the conversion process.

   Click **Finish**. The Required Test Fields dialog box opens and displays the missing required fields of the Cruise Booking test.

8. Specify the required test fields.

   a. Select the following:

      Level: Basic

      Priority: 4-Very High

      Reviewed: Reviewed

   b. Click **OK**. The Required Test Fields dialog box reopens and displays the missing required fields of the Cruise Search test.

   c. Select the same values entered for the Cruise Booking test. Click **OK**.

   d. Click **OK** to close the Convert to Tests wizard.

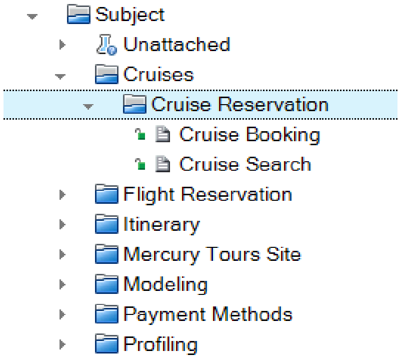
9. View the tests in the test plan tree.

   a. On the ALM sidebar, under **Testing**, select **Test Plan**.

   b. Choose **View > Test Plan Tree** to display the test plan tree.

   c. Expand **Cruises**. The test plan tree displays **Cruise Reservation** under **Cruises**.

d. Expand **Cruise Reservation**. The test plan tree displays the **Cruise Booking** and **Cruise Search** tests.



Planning Tests

After you define your requirements, you need to determine your testing goal and outline the strategy for achieving your goal.

After you determine your testing goal, you build a test plan tree, which hierarchically divides your application into testing units, or subjects. For each subject in the test plan tree, you define tests that contain steps. For each test step, you specify the actions to be performed on your application and the expected result.

ALM enables you to use the same test to test different use-cases, each with its own test configuration. Each test configuration uses a different set of data. You define the data by adding test parameter values for each test configuration. A test parameter is a variable that can be assigned a value.

When you create a test, a single test configuration with the same name as the test is created simultaneously. You can create as many additional test configurations as needed.

It is essential that the tests in your test plan comply with your requirements. To help ensure compliance throughout the application lifecycle management process, add coverage between tests and requirements. For finer granularity, you add coverage between test configurations and requirements.

In this lesson, you will learn about:

Developing a Test Plan Tree

The typical application is too large to test as a whole. The Test Plan module enables you to divide your application according to functionality. You divide your application into units, or subjects, by creating a test plan tree. The test plan tree is a graphical representation of your test plan, displaying your tests according to the hierarchical relationship of their functions. After you define subjects in the tree, you decide which tests to create for each subject, and add them to the tree.

In this exercise, you will add a subject and a test to the test plan tree in the Test Plan module.

To develop a test plan tree:

1. Open the ALM_Demo project.

   If the **ALM_Demo** project is not already open, log in to the project. For details, see "Start" on page 10.

2. Display the Test Plan module.

   On the ALM sidebar, under **Testing**, select **Test Plan**.

3. Add a subject folder to the test plan tree.

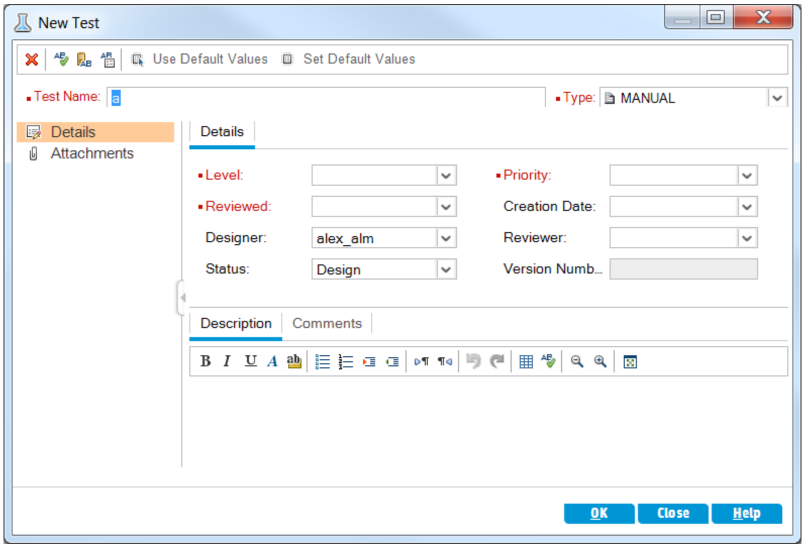
   a. Select the **Subject** folder and click the **New Folder** button. The New Test Folder dialog box opens.

   b. In the **Test Folder Name** box, type Payment Methods. Click **OK**. The new folder is added to the test plan tree.

   c. In the **Description** tab in the right pane, type a description of the subject: This folder contains tests that verify the payment methods.

4. Add a test to the subject folder.

   a. Select the Payment Methods folder and click the **New Test** button. The New Test dialog box opens.



Test Name: a

Type: MANUAL

Level:

Priority:

Reviewed:

Creation Date:

Designer: alex_alm

Reviewer:

Status: Design

Version Numb...

b. In the **Test Name** box, type a name for the test: Credit Cards.

c. In the **Type** box, select **MANUAL** to create a manual test.

d. In the **Details** tab, select the following:

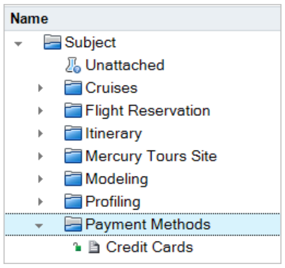
   Level: Basic

   Reviewed: Not Reviewed

   Priority: 4-Very High

e. In the **Description** tab, type a description for the test: The test verifies credit card types.

f. Click **OK**. The new test is added to the test plan tree under the Payment Methods folder.



Designing Test Steps

After you add a test to the test plan tree and define basic test information, you define test steps—detailed, step-by-step instructions that specify how to execute the test. A step includes the actions to be performed on your application and the expected results. You can create test steps for both manual and automated tests. For manual tests, you complete test planning by designing the test steps. Using your plan, you can begin test execution immediately. For automated tests, you create automated test scripts using OpenText testing tools, custom testing tools, or third-party testing tools.

In this exercise, you add test steps to the **Credit Cards** test. This test verifies the credit card type used to book a flight.

To design a test step:

1. Make sure the **Test Plan** module is displayed.

   If the Test Plan module is not displayed, on the ALM sidebar, under **Testing**, select **Test Plan**.

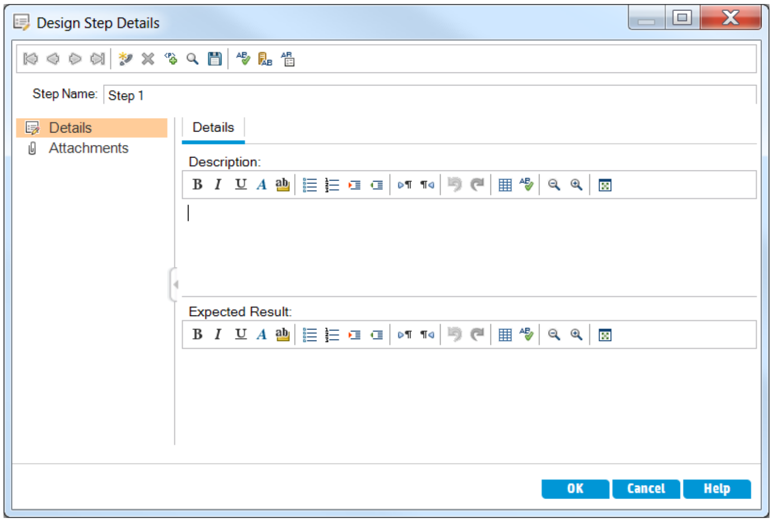
2. Display the **Credit Cards** test.

   Expand the **Payment Methods** folder, and select the **Credit Cards** test.

3. Open the Design Step Details dialog box.

   a. Click the **Design Steps** tab.

   b. Click the **New Step** button. The **Design Step Details** dialog box opens.



In the **Step Name** box, a step name is displayed. The default name is the sequential number of the test step.

4. Define the first test step.

   In the Design Step Details dialog box, type the following:

   Step Name: Step 1: Log in to Mercury Tours.

   Description:

   1. Enter URL.

   2. Log in.

   Expected Result: User is logged in to Mercury Tours.

5. Close the Design Step Details dialog box.

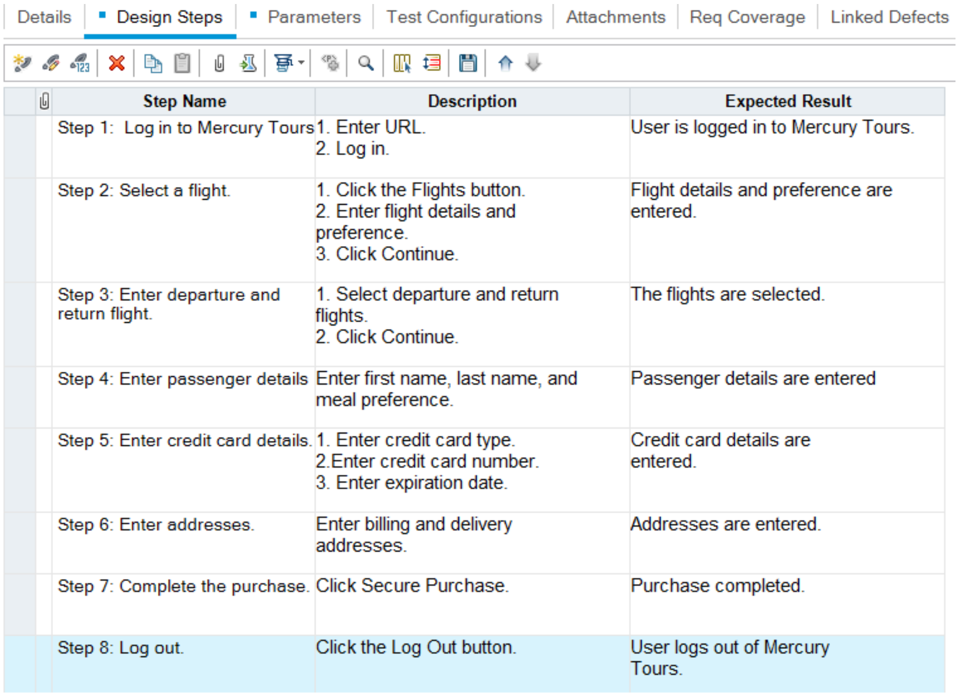
   Click **OK**.

6. Add the remaining test steps.

For each of the following test steps, click the **New Step** button to open the Design Step Details dialog box, type the required information, and click **OK** to close the Design Step Details dialog box:

Step Name	Description	Expected Result
Step 2: Select a flight destination.	1. Click the Flights button. 2. Enter flight details and preference. 3. Click Continue.	Flight details and preference are entered.
Step 3: Enter departure and return flight.	1. Select departure and return flights. 2. Click Continue.	The flights are selected.
Step 4: Enter passenger details.	Enter first name, last name, and meal preference.	Passenger details are entered.
Step 5: Enter credit card details.	1. Enter credit card type. 2. Enter credit card number. 3. Enter expiration date.	Credit card details are entered.
Step 6: Enter addresses.	Enter billing and delivery addresses.	Addresses are entered.
Step 7: Complete the purchase.	Click Secure Purchase.	Purchase completed.
Step 8: Log out.	Click the Log Out button.	User logs out of Mercury Tours.

The Design Steps tab displays the design steps.



Step Name	Description	Expected Result
Step 1: Log in to Mercury Tours	1. Enter URL. 2. Log in.	User is logged in to Mercury Tours.
Step 2: Select a flight.	1. Click the Flights button. 2. Enter flight details and preference. 3. Click Continue.	Flight details and preference are entered.
Step 3: Enter departure and return flight.	1. Select departure and return flights. 2. Click Continue.	The flights are selected.
Step 4: Enter passenger details	Enter first name, last name, and meal preference.	Passenger details are entered
Step 5: Enter credit card details.	1. Enter credit card type. 2.Enter credit card number. 3. Enter expiration date.	Credit card details are entered.
Step 6: Enter addresses.	Enter billing and delivery addresses.	Addresses are entered.
Step 7: Complete the purchase.	Click Secure Purchase.	Purchase completed.
Step 8: Log out.	Click the Log Out button.	User logs out of Mercury Tours.

Defining Test Parameters

To increase the flexibility of your tests, you can add parameters to your tests. This enables you to run the same test repeatedly with different data each time.

When working with a manual test, you can add parameters to the design steps from within the test or you can add parameters by calling them from other tests. This is useful if you have common steps you often want to perform as part of other tests.

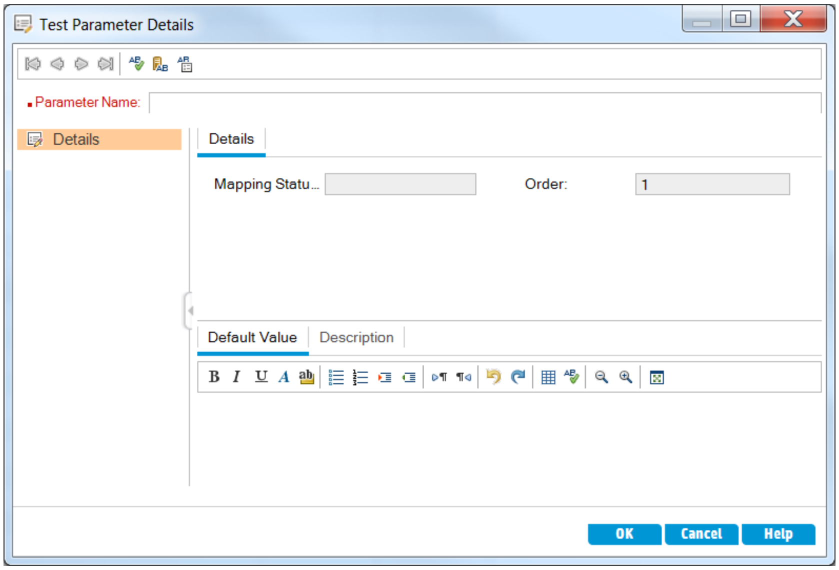
When working with an automated test, you can define parameters for a test script from within the test or you can load parameters from a shared test resource file.

When defining a test configuration, you define data by setting test parameter values for each test configuration.

In "Designing Test Steps" on page 33, you defined steps for the **Credit Cards** test. In this exercise, you will add parameters to enhance this test.

To define test parameters:

1. Display the Parameters tab for the Credit Cards test.
   a. In the test plan tree, expand the **Payment Methods** folder, and select the **Credit Cards** test.
   b. Click the **Parameters** tab.
2. Add a parameter.
   a. Click the **New Parameter** button. The Test Parameter Details dialog box opens.


Test Parameter Details

Parameter Name:

Mapping Statu...

Order: 1

   b. Type the following:

      Parameter Name: Credit card type.

      Default Value: American Express, Visa, or MasterCard.
   c. Click **OK** to close the Test Parameter Details dialog box. The parameter is added to the Parameters tab.
3. Add an additional parameter.
   a. Click the **New Parameter** button. The Test Parameter Details dialog box opens.

b. Type the following:

Parameter Name: Credit card number.

Default Value:1111-2222-3333-4444.

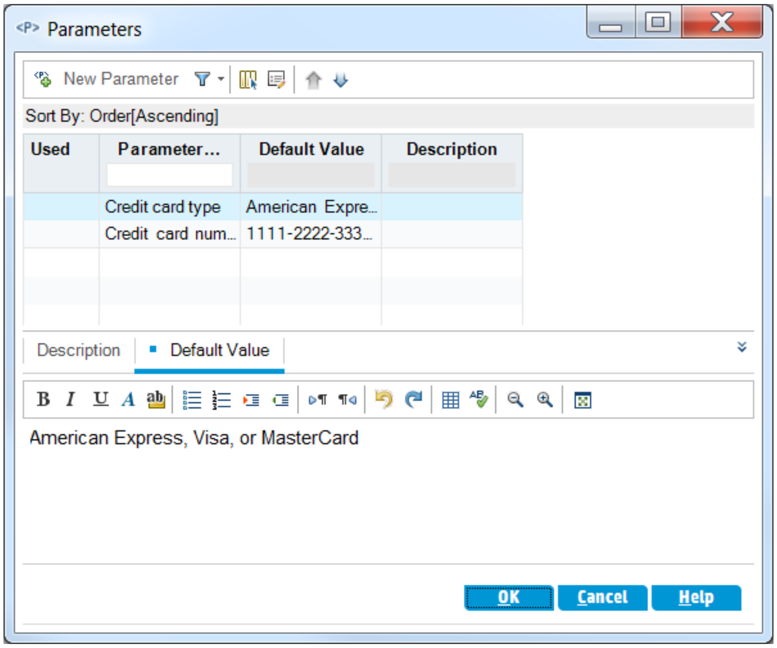
c. Click **OK** to close the Test Parameter Details dialog box. The parameter is added to the Parameters tab.

4. Assign parameters to the test steps.

a. Click the **Design Steps** tab.

b. Click the **Description** box of Step 5.

c. Place the cursor after 1. Enter credit card type and click the **Insert Parameter** button. The Parameters dialog box opens.



Sort By: Order[Ascending]

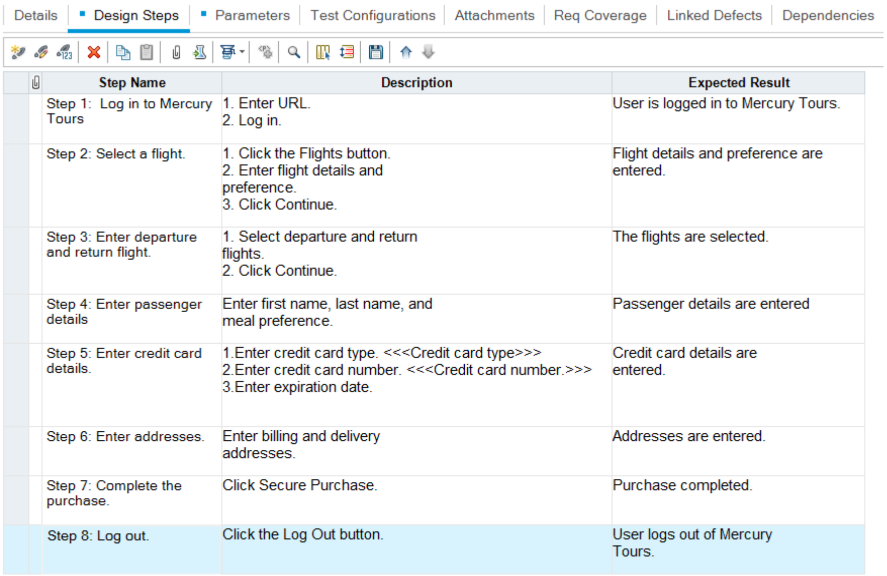
Used	Parameter...	Default Value	Description
	Credit card type	American Expre...	
	Credit card num...	1111-2222-333...	

American Express, Visa, or MasterCard

d. Select the **Credit Card Type** parameter. Click OK.

e. Place the cursor after 2. Enter credit card number and click the **Insert Parameter** button. The Parameters dialog box opens. Select the **Credit Card Number** parameter. Click OK.

f. The parameters are added to your design step.



Step Name	Description	Expected Result
Step 1: Log in to Mercury Tours	1. Enter URL. 2. Log in.	User is logged in to Mercury Tours.
Step 2: Select a flight.	1. Click the Flights button. 2. Enter flight details and preference. 3. Click Continue.	Flight details and preference are entered.
Step 3: Enter departure and return flight.	1. Select departure and return flights. 2. Click Continue.	The flights are selected.
Step 4: Enter passenger details	Enter first name, last name, and meal preference.	Passenger details are entered
Step 5: Enter credit card details.	1.Enter credit card type. <<<Credit card type>>> 2.Enter credit card number. <<<Credit card number.>>> 3.Enter expiration date.	Credit card details are entered.
Step 6: Enter addresses.	Enter billing and delivery addresses.	Addresses are entered.
Step 7: Complete the purchase.	Click Secure Purchase.	Purchase completed.
Step 8: Log out.	Click the Log Out button.	User logs out of Mercury Tours.

Defining Test Configurations

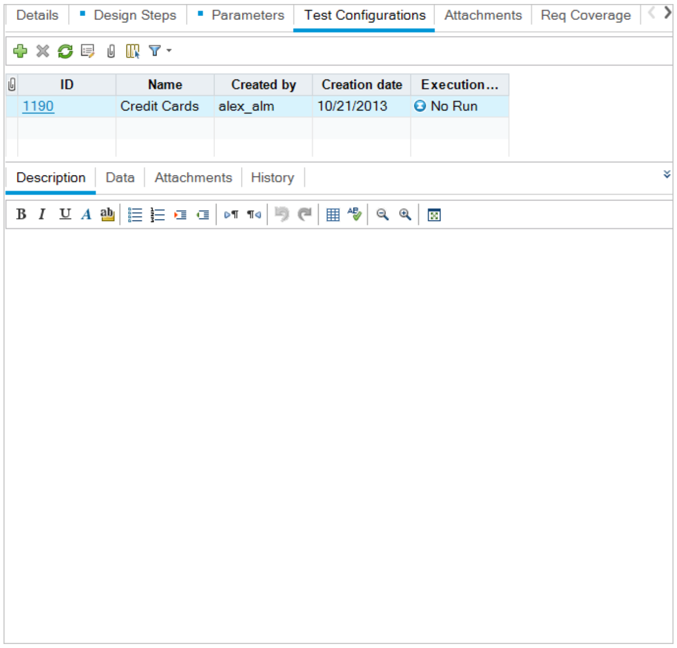
You can reuse a test to test different use-cases. For each use-case you create a test configuration that uses a different data set. When working with a manual test, the data set of a test configuration contains values for your defined test parameters. When working with a UFT or a business process test, the data set can use an external test resource file.

In the following exercise, you will create a test configuration for each of the following credit cards: American Express, Visa, and MasterCard. You will then define the actual parameter values to be used when running these instances.

To define test configurations:

1. Display the Test Configurations tab for the Credit Cards test.
   a. In the test plan tree, expand the **Payment Methods** folder, and select the **Credit Cards** test.

b. Click the **Test Configurations** tab.



ID	Name	Created by	Creation date	Execution...
1190	Credit Cards	alex_alm	10/21/2013	No Run

By default, ALM created the Credit Cards test configuration.

2. Rename the test configuration name to American Express.

   a. Under **Name**, click **Credit Cards**. Type **American Express**.

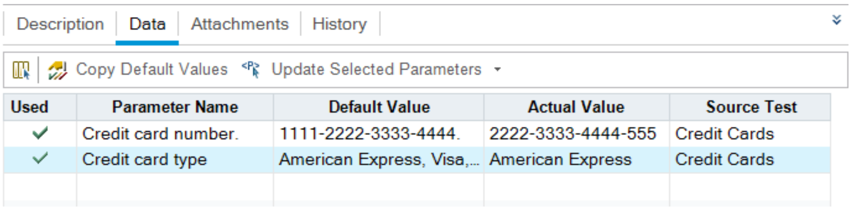
   b. Under **Description**, type: Test configuration for American Express.

3. Assign data to the test configuration.

   a. Click the **Data** tab.

   b. Under **Actual Value**, click the top cell. Click the arrow and click in the box. Type: 2222-3333-4444-5555. Click **OK**.

   c. Under **Actual Value**, click the second cell. Click the arrow and click in the box. Type: American Express. Click **OK**.



Used	Parameter Name	Default Value	Actual Value	Source Test
✓	Credit card number.	1111-2222-3333-4444.	2222-3333-4444-555	Credit Cards
✓	Credit card type	American Express, Visa,...	American Express	Credit Cards

4. Add a new test configuration for Visa.

   a. Click the **New Test Configuration** button. The New Test Configuration dialog box opens.
   b. Type the following:

      Name: Visa

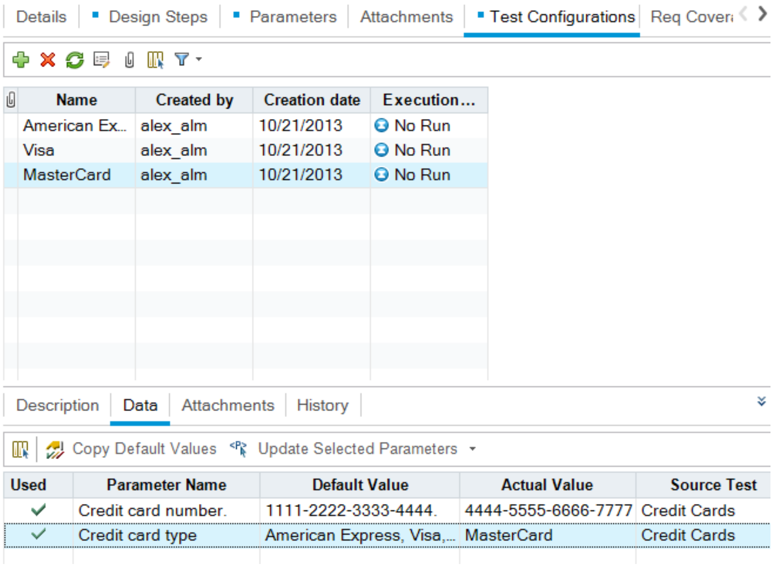
      Description: Test configuration for Visa.
   c. Click **OK**. The test configuration is added to the Test Configuration tab.
   d. Make sure the Visa test configuration is selected.
   e. Click the **Data** tab. Under **Actual Value**, click the top cell. Click the arrow and click in the box. Type: 3333-4444-5555-6666. Click **OK**.
   f. Under **Actual Value**, click the second cell. Click the arrow and click in the box. Type: Visa. Click **OK**.

5. Add a new test configuration for MasterCard.
   a. Click the **New Test Configuration** button. The New Test Configuration dialog box opens.
   b. Type the following:

      Name: MasterCard

      Description: Test configuration for MasterCard.
   c. Click **OK**. The test configuration is added to the Test Configuration tab.
   d. Make sure the MasterCard test configuration is selected.
   e. Click the **Data** tab. Under **Actual Value**, click the top cell. Click the arrow and click in the box. Type: 4444-5555-6666-7777. Click **OK**.

f. Under **Actual Value**, click the second cell. Click the arrow and click in the box. Type: MasterCard. Click **OK**.



Name	Created by	Creation date	Execution...
American Ex...	alex_alm	10/21/2013	No Run
Visa	alex_alm	10/21/2013	No Run
MasterCard	alex_alm	10/21/2013	No Run

Used	Parameter Name	Default Value	Actual Value	Source Test
✓	Credit card number.	1111-2222-3333-4444.	4444-5555-6666-7777	Credit Cards
✓	Credit card type	American Express, Visa,...	MasterCard	Credit Cards

6. **Version Control**: Check in the test configurations.

   Check in the test and its test configurations. In the test plan tree, right-click the **Credit Cards** test, and select **Versions > Check In**. Click **OK** to confirm.

Creating and Viewing Coverage

It is essential that the tests in your test plan comply with your requirements. To help ensure compliance throughout the application lifecycle management process, you can add coverage between your tests and requirements. You can also add coverage between test configurations and requirements.

You can create coverage from the Test Plan module and the Requirements module. A test or a test configuration can cover more than one requirement, and a requirement can be covered by more than one test or test configuration.

In these exercises, you will learn about the following:

Creating Coverage

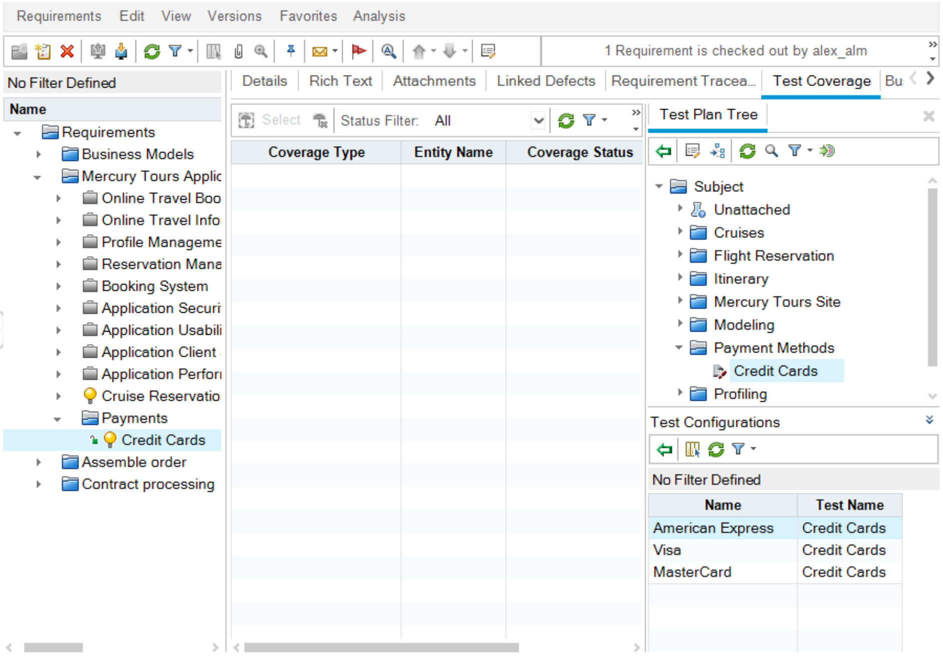
In this exercise, you will create the **Credit Cards** requirement and then create coverage by associating it to the **Credit Cards** test.

To create coverage:

1. Display the Requirements module.
   a. On the ALM sidebar, under **Requirements**, select **Requirements**.
   b. Choose **View > Requirement Details**. The Requirement Details view is displayed.
2. Create the Credit Card requirement.
   a. Select the **Mercury Tours Application** folder.
   b. Click the **New Folder** button. In the New Requirement Folder dialog box, type: Payments. Click **OK**.
   c. Select the **Payments** folder and click the **New Requirement** button. The New Requirement dialog box opens.
   d. In the **Name** box, type Credit Cards.
   e. In the **Requirement Type** box, select **Functional**.
   f. Click **Submit**. Click **Close**. The new requirement is added to the requirements tree.
3. Display the Test Coverage tab.
   a. In the requirements tree, make sure that the **Credit Cards** requirement is selected.
   b. In the right pane, click the **Test Coverage** tab.

4. Display the Test Plan Tree pane.

   Click the **Select Tests** button to show the test plan tree on the right.

Name	Test Name
American Express	Credit Cards
Visa	Credit Cards
MasterCard	Credit Cards

5. Select the Credit Cards test in the test plan tree.

   a. In the Test Plan Tree pane, expand the **Payment Methods** folder, and select the **Credit Cards** test.

   b. If the Test Configurations pane is not displayed, click the **Show** button on the bottom of the pane. Under the Test Configurations pane, you can see that the test contains three test configurations.

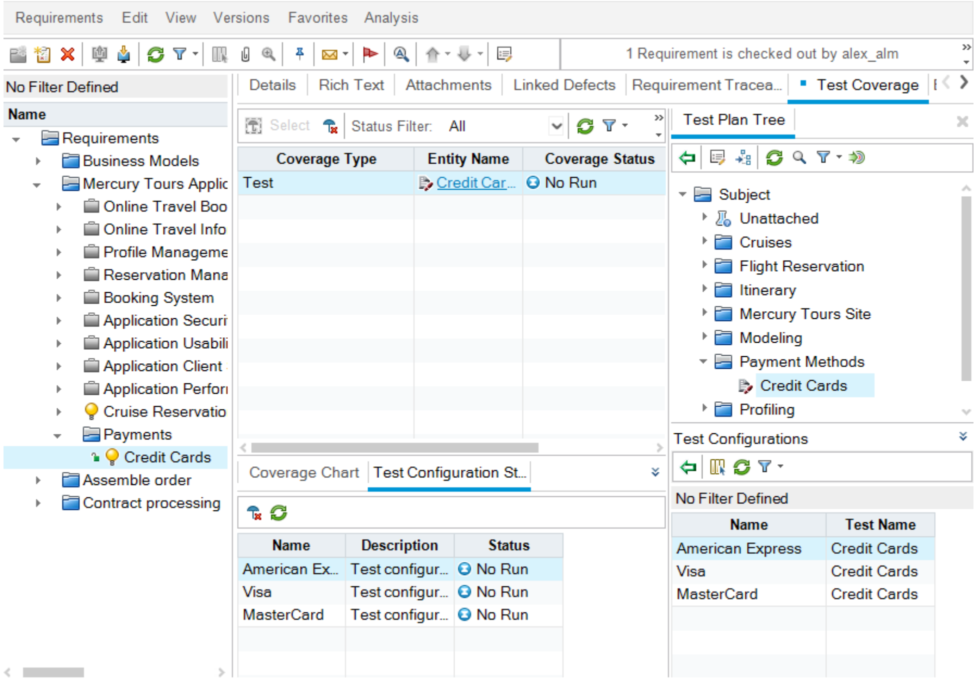
6. Add the test to the coverage grid.

   In the Test Plan Tree pane, click the **Add to Coverage** button. The **Credit Cards** test is added to the coverage grid.

7. Display the Test Configuration Status tab.

   Click the **Test Configuration Status** tab.

The Test Configuration Status tab shows the associated test configurations and their status.



1 Requirement is checked out by alex_alm

Coverage Type	Entity Name	Coverage Status
Test	Credit Car...	No Run

Name	Description	Status
American Ex...	Test configur...	No Run
Visa	Test configur...	No Run
MasterCard	Test configur...	No Run

Name	Test Name
American Express	Credit Cards
Visa	Credit Cards
MasterCard	Credit Cards

> **Tip:** To add selected configurations of a test to the requirement's test coverage, add coverage from the Test Configurations pane.

8. Hide the test plan tree.

   Click the **Close** button above the tests plan tree.

9. **Version Control**: Check in the Payments folder and the Credit Cards requirement.

   a. In the test plan tree, right-click the **Payments** folder, and select **Versions > Check In**. Click **OK** to confirm.

   b. Right-click the **Credit Cards** requirement, and select **Versions > Check In**. Click **OK** to confirm.

Analyzing Coverage

After you create test coverage, you can use the Coverage Analysis view in the Requirements module to analyze the breakdown of child requirements according to test coverage.

In this exercise, you will analyze the **Application Client System** requirement.

To analyze test coverage:

1. Make sure that the Requirements module is displayed.

   If the Requirements module is not displayed, on the ALM sidebar, under **Requirements**, select **Requirements**.

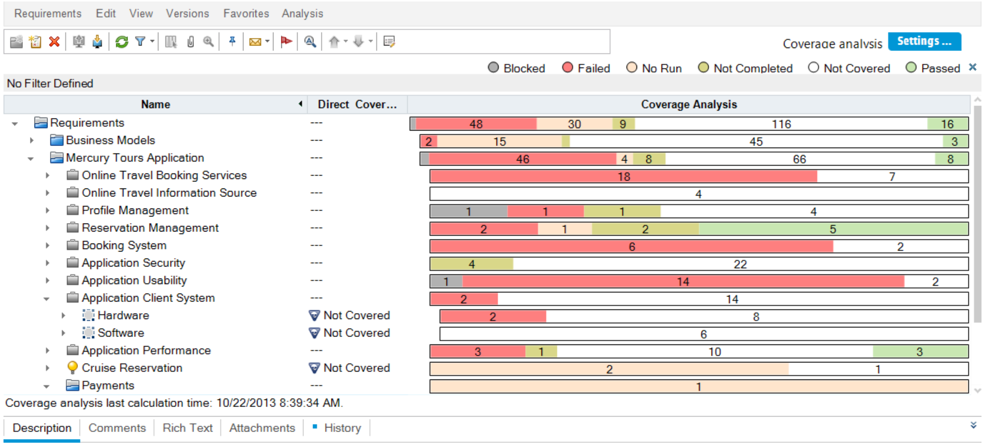
2. Display the requirements tree in Coverage Analysis view.

   Choose **View > Coverage Analysis**. The Coverage Analysis view is displayed.

3. Display the Application Client System requirement in Coverage Analysis view.

   a. If any filters are applied, click the **Filter** arrow and choose **Clear Filter/Sort**. Click **Yes** to confirm.

   b. Under the **Mercury Tours Application** requirement, expand the **Application Client System** requirement and its children.



No Filter Defined

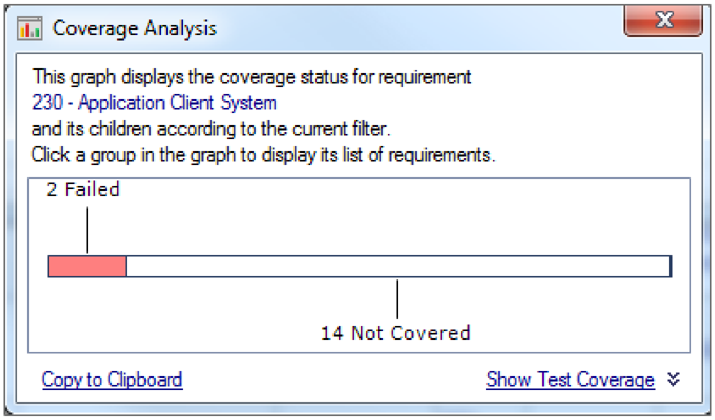
Name	Direct Cover...	Coverage Analysis
Requirements	---	48 30 9 116 16
Business Models	---	2 15 45 3
Mercury Tours Application	---	46 4 8 66 8
Online Travel Booking Services	---	18 7
Online Travel Information Source	---	4
Profile Management	---	1 1 1 4
Reservation Management	---	2 1 2 5
Booking System	---	6 2
Application Security	---	4 22
Application Usability	---	1 14 2
Application Client System	---	2 14
Hardware	Not Covered	2 8
Software	Not Covered	6
Application Performance	---	3 1 10 3
Cruise Reservation	Not Covered	2 1
Payments	---	1

Coverage analysis last calculation time: 10/22/2013 8:39:34 AM.

   In the Coverage Analysis column, you can see graphically the number of child requirements that have a direct cover status and those that are not yet covered.

4. Display coverage analysis for the Application Client System requirement.

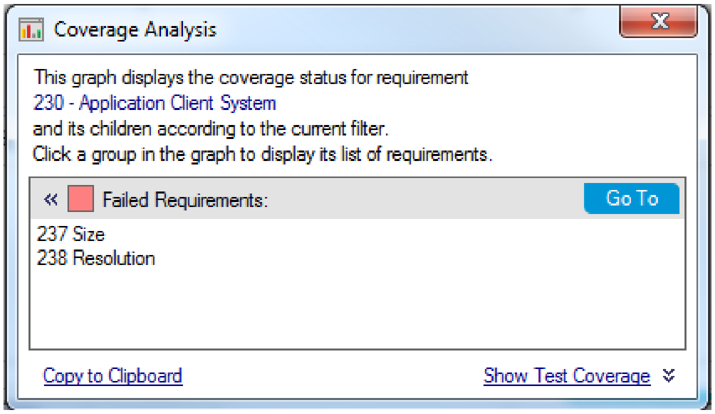
Right-click the **Application Client System** requirement, and choose **Coverage Analysis**. The Coverage Analysis dialog box opens.



This graph displays the coverage status for requirement
230 - Application Client System
and its children according to the current filter.
Click a group in the graph to display its list of requirements.

5. Display the child requirements with a “Failed” status.

   Click the red **Failed** area of the graph. The child requirements with a “Failed” status are listed....



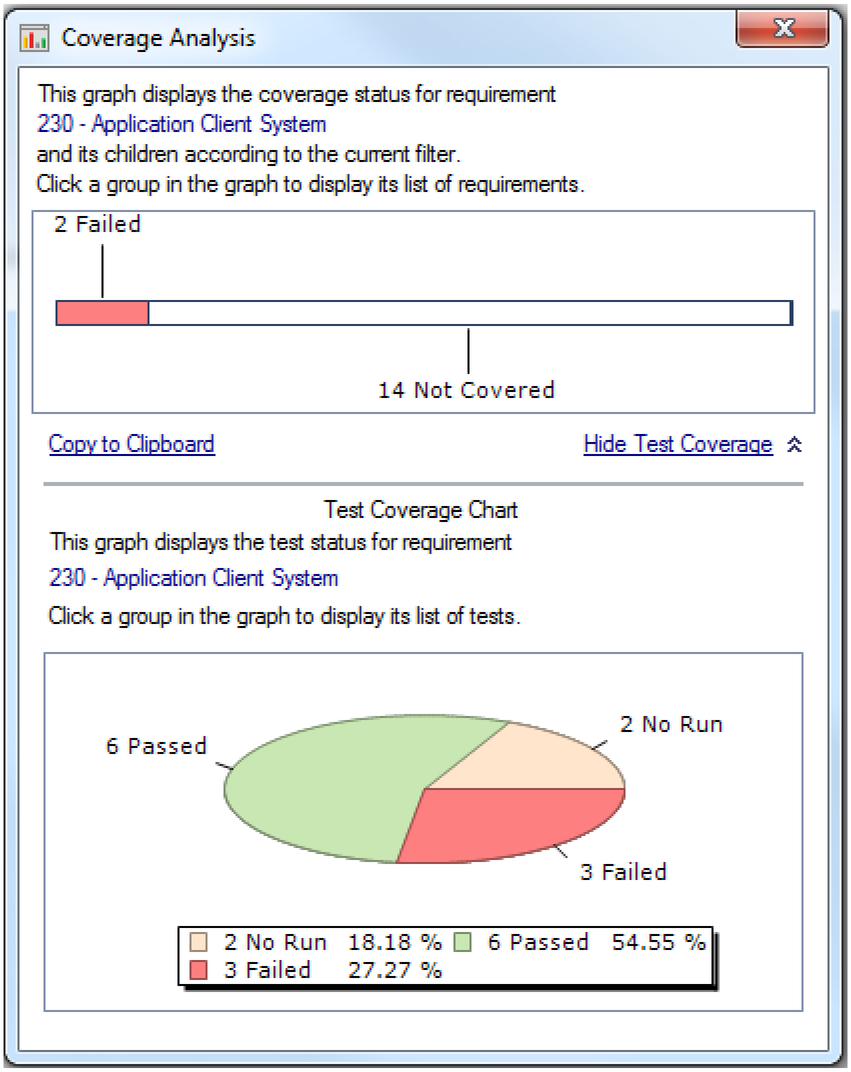
This graph displays the coverage status for requirement
230 - Application Client System
and its children according to the current filter.
Click a group in the graph to display its list of requirements.

Failed Requirements:

237 Size
238 Resolution

6. Display test coverage for the requirement.

   a. Click the **Show Test Coverage** link to extend the Coverage Analysis dialog box and display the Test Coverage Chart.



This graph displays the coverage status for requirement
230 - Application Client System
and its children according to the current filter.
Click a group in the graph to display its list of requirements.

Test Coverage Chart
This graph displays the test status for requirement
230 - Application Client System
Click a group in the graph to display its list of tests.

This pie chart graphically displays the full test coverage for the requirement, grouped according to test status.

b. Click the **Passed** section of the chart to open the Tests Coverage dialog box and display the list of tests with the selected status. Close the Test Coverage dialog box.

7. Close the Coverage Analysis dialog box.

   Click the **Close** button.

Copying Test Steps

You can copy steps from another test in the same project or from a different project. In this exercise, you will copy the test steps from the **HTML Page Layout** test and paste them into a newly created test.

To copy a test step:

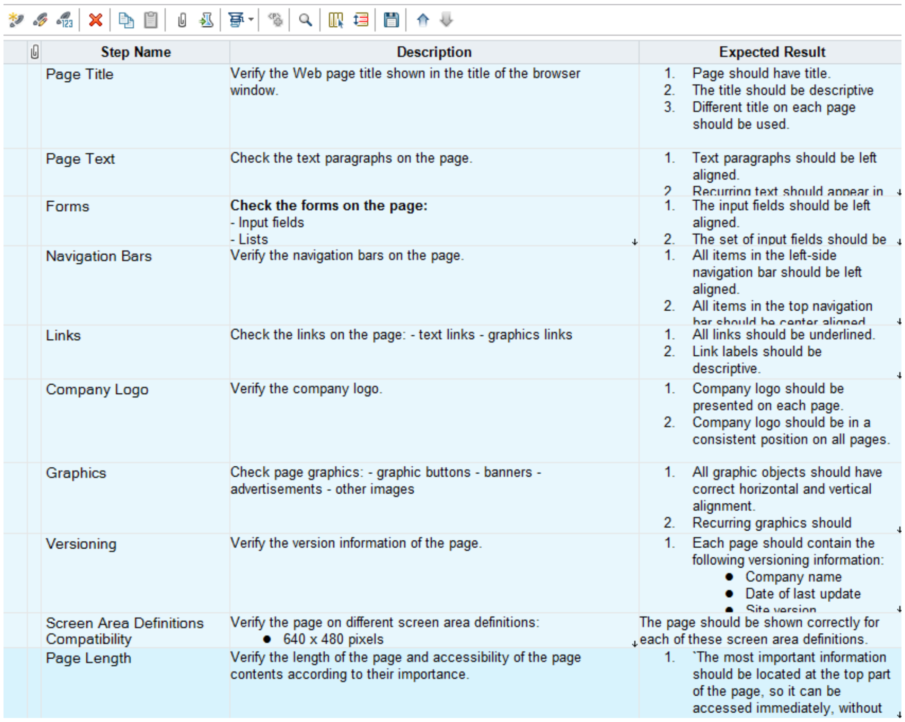
1. Display the **Test Plan** module.
   a. On the ALM sidebar, under **Testing**, select **Test Plan**.
   b. If the test plan tree view is not displayed, select **View > Test Plan Tree**.
2. Create a new test.
   a. In the test plan tree, expand the **Mercury Tours Site** folder.
   b. Select the **HTML Pages** folder and click the **New Test** button. The New Test dialog box opens.
   c. In the **Test Name** box, type a name for the test: New HTML Page Layout.
   d. In the **Type** box, select **MANUAL** to create a manual test.
   e. In the Details tab, select the following:

      Level: Basic

      Reviewed: Not Reviewed

      Priority: 4-Very High
   f. Click **OK**. The new test is added to the test plan tree under the **HTML Pages** folder.
3. Display the Design Steps tab for the HTML Page Layout test.
   a. In the **HTML Pages** folder, select the **HTML Page Layout** test.
   b. Click the **Design Steps** tab.
4. Select the steps that you want to copy.

Position the mouse pointer in the gray sidebar on the left. The mouse pointer changes to a pointing hand. Select all rows.



Step Name	Description	Expected Result
Page Title	Verify the Web page title shown in the title of the browser window.	1. Page should have title. 2. The title should be descriptive 3. Different title on each page should be used.
Page Text	Check the text paragraphs on the page.	1. Text paragraphs should be left aligned. 2. Recurring text should appear in
Forms	**Check the forms on the page:** - Input fields - Lists	1. The input fields should be left aligned. 2. The set of input fields should be
Navigation Bars	Verify the navigation bars on the page.	1. All items in the left-side navigation bar should be left aligned. 2. All items in the top navigation bar should be center aligned
Links	Check the links on the page: - text links - graphics links	1. All links should be underlined. 2. Link labels should be descriptive.
Company Logo	Verify the company logo.	1. Company logo should be presented on each page. 2. Company logo should be in a consistent position on all pages.
Graphics	Check page graphics: - graphic buttons - banners - advertisements - other images	1. All graphic objects should have correct horizontal and vertical alignment. 2. Recurring graphics should
Versioning	Verify the version information of the page.	1. Each page should contain the following versioning information: ● Company name ● Date of last update ● Site version
Screen Area Definitions Compatibility	Verify the page on different screen area definitions: ● 640 x 480 pixels	The page should be shown correctly for each of these screen area definitions.
Page Length	Verify the length of the page and accessibility of the page contents according to their importance.	1. `The most important information should be located at the top part of the page, so it can be accessed immediately, without

5. Copy the selected steps.

   Click the **Copy Steps** button.

6. Paste the steps into the New HTML Page Layout test.

   a. In the test plan tree, select the **New HTML Page Layout** test.

   b. In the **Design Steps** tab, click the **Paste Steps** button. The test steps are copied to the Design Steps tab.

Generating Automated Test Scripts

Test planning involves deciding which tests to automate. If you choose to execute tests manually, the tests are ready for execution as soon as you define the test steps. If you choose to automate tests, you can generate test scripts and complete them using other OpenText testing tools (for example, UFT One).

Consider these issues when deciding whether to automate a test.

Do Automate	Do Not Automate
Tests that run with each new version of your application to check the stability of basic functionality across the entire application (regression tests).	Tests that are executed only once.
Tests that use multiple data values for the same operation (data-driven tests).	Tests that require immediate execution.
Tests that are run many times (stress tests) and tests that check a multi-user client/server system (load tests).	Tests that check how easy the application is to use (usability tests).
	Tests that do not have predictable results.

In this exercise, you will generate a UFT test script for the **Address Options** test.

> **Note:** For prerequisites to working with a UFT test, see "Tutorial " on page 6.

To generate an automated test script:

1. Make sure the test plan tree view is displayed.

   If the test plan tree view is not displayed, select **View > Test Plan Tree**.

2. Locate the Address Options manual test.

   a. Select the **Subject** folder at the root of the test plan tree and choose **Edit > Find**. The Find dialog box opens.

   b. In **Value To Find**, type Book.

   c. In the **Search for**, select **Folders**.

   d. Click **Find**. The Search Results dialog box opens and displays a list of possible matches.

   e. Double-click the **Flight Reservation\Book Flight** folder to highlight the folder in the test plan tree. Click **Close** to close the Search Results dialog box.

   f. In the test plan tree, expand the **Book Flight** folder and select the **Address Options** test.

3. Display the Design Steps tab.

   In the right pane, click the **Design Steps** tab.

4. Generate a test script.

   a. Click the **Generate Script** button.

   b. Choose **QUICKTEST_TEST** to generate a UFT test.

   c. **Version Control**: If a check out message box opens, click **OK**.

   The steps in the **Address Options** test are used to create the automated test script.

5. View the test script.

   a. Click the **Test Script** tab.

   b. To display and modify your test script in UFT, click the **Launch OpenText Functional Testing** button.

Running Tests

Throughout the application lifecycle management process, you can run automated and manual tests to locate defects and assess the quality of your application.

You start by creating **test sets** and choosing which tests to include in each set. A test set contains a subset of the tests in an ALM project designed to achieve specific test goals.

After you define test sets, you can begin to execute your tests. Some tests can be run automatically and some can be run manually.

When you run a test automatically, ALM opens the selected testing tool, which runs the test, and imports the test results to ALM.

When you run a test manually, you execute the test steps you defined in test planning. You pass or fail each step, depending on whether the actual results match the expected output.

If you are using **ALM Edition** with the Lab Management extension enabled, you can use server-side execution to reserve testing resources for automated tests. Server-side execution occurs on remote testing hosts, can be scheduled or immediate, and does not require user intervention.

ALM enables you to control the execution of tests in a test set by setting conditions and scheduling the date and time for executing your tests.

After test execution, you can use ALM to view and analyze the results of your tests.

In this lesson, you will learn about:

Test Set Types

After you design tests in the Test Plan module, you create a test sets tree in the Test Lab module. A test sets tree enables you to organize your testing needs by grouping test sets in folders and organizing them in different hierarchical levels in the Test Lab module. You assign each test set folder to a cycle. This enables you to group together test sets that will be run during the same cycle and analyze the progress of the cycle as you run your tests.

When defining a test set, you add instances of your selected tests to the test set. Each test instance contains a defined test configuration.

ALM provides the following types of test sets:

- **Functional** test sets include automatic tests that check the application under test functions as expected. Tests in a Functional test set are scheduled in a timeslot to run on a server, without requiring user supervision. **Available for**: ALM Edition with the Lab Management Extension enabled.
- **Default** test sets can include automatic and manual tests, and are used to check that the application under test functions as expected. Tests in a Default test set are controlled from the user’s machine and require the supervision of the tester.
- **Performance** test sets include performance tests which check that the application under test can withstand load and demand. Tests in a Performance test set are scheduled in a timeslot to run on a server, without requiring user supervision. **Available for**: ALM Edition and LoadRunner Enterprise Edition only.

> **Note:** For the purposes of this tutorial, we will only use Functional and Default test sets. The usage of Performance tests is covered in the LoadRunner Enterprise help.

To decide which types of test sets to create, consider the goals you defined at the beginning of the application lifecycle management process.

When creating and combining different groups of test sets, consider issues such as the current state of the application and the addition or modification of new features. Following are examples of general categories of test sets you can create:

Test Set	Description
Sanity	Checks entire application at a basic level—focusing on breadth, rather than depth—to verify that the application is functional and stable. This set includes fundamental tests that contain positive checks, validating that the application is functioning properly. For example, in the Mercury Tours application, you could test whether the application opens and enables you to log in.
Regression	Tests the system in a more in-depth manner than a sanity set. This set can include both positive and negative checks. Negative tests attempt to fail an application to demonstrate that the application is not functioning properly.
Advanced	Tests both breadth and depth. This set covers the entire application, and also tests the application’s advanced options. You can run this set when there is ample time for testing.
Function	Tests a subsystem of an application. This could be a single feature or a group of features. For example, in the Mercury Tours application, a function set could test all activities related to booking a flight.

Defining Test Sets

In this exercise, you will define the Mercury Tours Site test set. You will also set failure rules for the test set to instruct ALM how to proceed in the event that an automated test in the test set fails. Depending on whether you are an ALM Edition user, you can either define a Functional test set or a Default test set.

> **Note:**
>
> - If you are using ALM Edition with the Lab Management extension enabled, define a Functional test set. See "Defining a Functional Test Set" below.
> - If you are not using ALM Edition with Lab Management, define a Default test set. See "Defining a Default Test Set" on page 59.

Defining a Functional Test Set

Functional test sets contain automatic tests. Tests in a Functional test set run using automated server-side execution.

To define a Functional test set:

1. Display the **Test Lab** module.

   On the ALM sidebar, under **Testing**, select **Test Lab**.

2. Add a folder to the test sets tree.

   a. In the test sets tree in the left pane, select the **Root** folder.

   b. Click the **New Folder** button. The New Test Set Folder dialog box opens.

   c. In the **Folder Name** box, type Service Pack 1 and click **OK**.

3. Create subfolders for the test set folder.

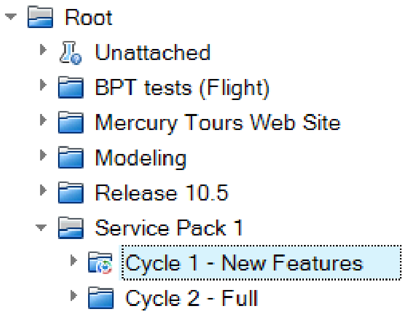
   Select the **Service Pack 1** folder and repeat the previous step to create two subfolders, named Cycle 1 - New Features, and Cycle 2 - Full.

4. Assign the test set folders to a cycle.

   a. Right-click the **Cycle 1 - New Features** test set folder and select the **Assign to Cycle** button. The Select Cycles dialog box opens.

   b. Expand the **Service Packs** releases folder. In the **Service Pack 1** release, select the **Cycle 1 - New Features** cycle (created in Lesson 2, "Specify releases and cycles" on page 15).

c. Click **OK**. The icon for the folder in the test sets tree changes to show that the folder has been assigned to a cycle.



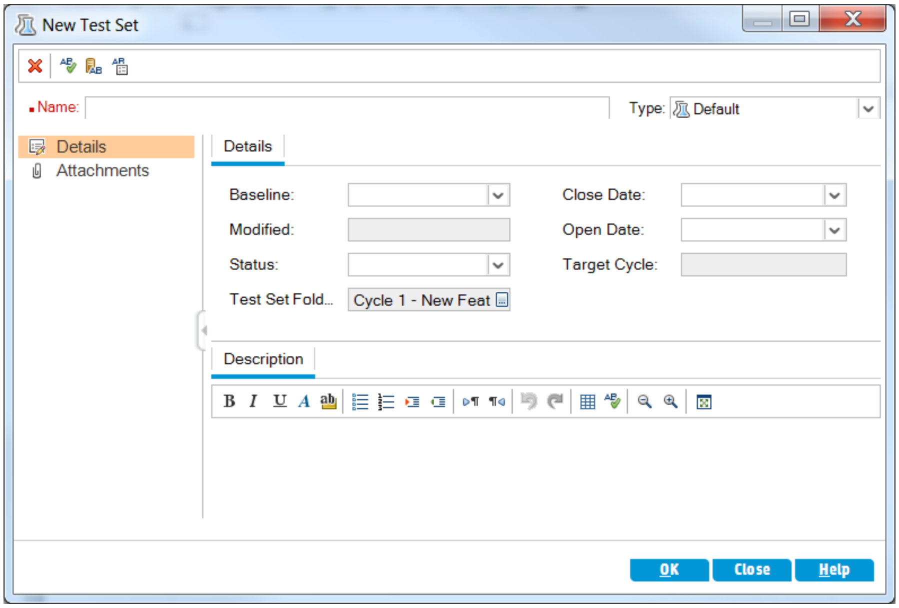
- Root
  - Unattached
  - BPT tests (Flight)
  - Mercury Tours Web Site
  - Modeling
  - Release 10.5
  - Service Pack 1
    - Cycle 1 - New Features
    - Cycle 2 - Full

d. Right-click the **Cycle 2 - Full** test sets folder and choose **Assign to Cycle**. Assign the folder to the **Cycle 2 - Full** cycle, located in the **Service Pack 1** release in the releases tree.

5. Add a test set to the Cycle 1 - New Features test set folder.

a. In the test sets tree, select **Cycle 1- New Features**.

b. Click the **New Test Set** button. The New Test Set dialog box opens.



New Test Set

Name:

Type: Default

Details

Attachments

Details

Baseline:

Modified:

Status:

Test Set Fold... Cycle 1 - New Feat

Close Date:

Open Date:

Target Cycle:

Description

c. Enter the following:

Name: Mercury Tours Site

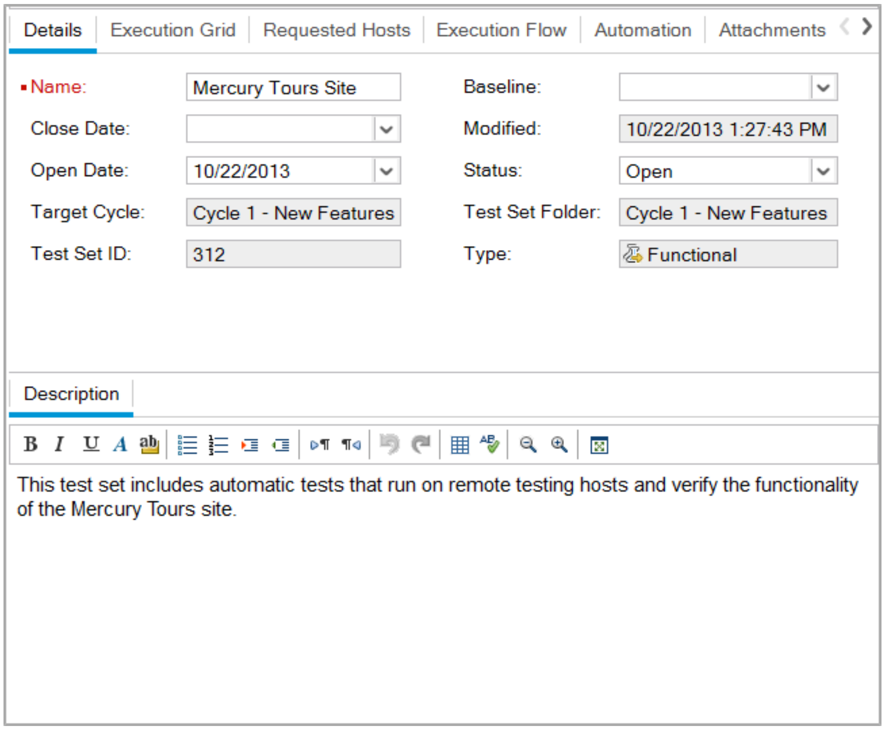
Description: This test set includes automatic tests that run on remote testing hosts and verify the functionality of the Mercury Tours site.

d. Select **Functional** in the **Type** field.

e. Click **OK**. The **Mercury Tours Site** test set is added to the test sets tree in the left pane.

6. Define the Mercury Tours Site test set details.

   a. Click the test set in the test sets tree. The **Execution Grid** tab is displayed. Click on the **Details** tab.



▪Name:	Mercury Tours Site	Baseline:	
Close Date:		Modified:	10/22/2013 1:27:43 PM
Open Date:	10/22/2013	Status:	Open
Target Cycle:	Cycle 1 - New Features	Test Set Folder:	Cycle 1 - New Features
Test Set ID:	312	Type:	Functional

This test set includes automatic tests that run on remote testing hosts and verify the functionality of the Mercury Tours site.

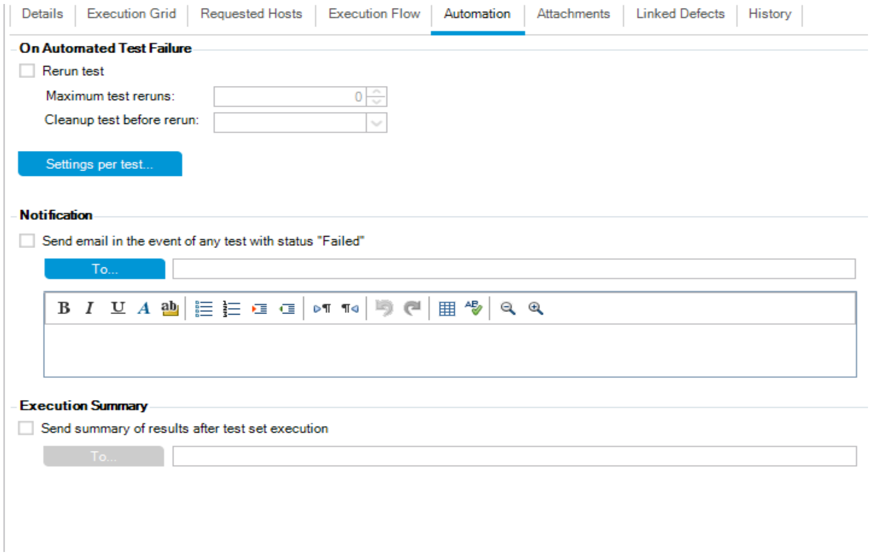
   b. Select the following:

   Open Date: Select a date from the calendar for the planned opening date for the test set. Today's date is selected by default.

   Close Date: Select the planned closing date for the test set.

7. Set rules for the automated tests in the test set in the event of a test failure.

a. Click the **Automation** tab.



On Automated Test Failure

□ Rerun test

Maximum test reruns: 0

Cleanup test before rerun:

Notification

□ Send email in the event of any test with status "Failed"

Execution Summary

□ Send summary of results after test set execution

b. In the **On Automatic Test Failure** section, perform the following:

- Select the **Rerun test** check box.
- Set **Maximum test reruns** to **1**.

8. Instruct ALM to send an email to specific users if certain events occur.

   Under **Notification**, perform the following:

   a. Select the check box to send an email notification if any test in the test set fails.

   b. **To**: Enter your email address.

   c. **Message**: Type the following:

   This test failed. Please review the test results and submit a defect.

Defining a Default Test Set

Default test sets contain automatic and manual tests. You start and control tests in a Default test set using your local machine.

> **Note:** In this exercise, you define a Default test set. If you are using ALM Edition with Lab Management enabled, you already defined a Functional test set. You can proceed to "Adding Tests to a Test Set" on page 64.

To define a Default test set:

1. Display the **Test Lab** module.

   On the ALM sidebar, under **Testing**, select **Test Lab**.

2. Add a folder to the test sets tree.

   a. In the test sets tree in the left pane, select the **Root** folder.

   b. Click the **New Folder** button. The New Test Set Folder dialog box opens.

   c. In the **Folder Name** box, type Service Pack 1 and click **OK**.

3. Create subfolders for the test set folder.

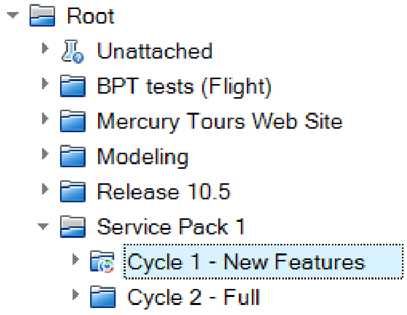
   Select the **Service Pack 1** folder and repeat the previous step to create two subfolders, named Cycle 1 - New Features, and Cycle 2 - Full.

4. Assign the test set folders to a cycle.

   a. Right-click the **Cycle 1 - New Features** test set folder and select the **Assign to Cycle** button. The Select Cycles dialog box opens.

   b. Expand the **Service Packs** releases folder. In the **Service Pack 1** release, select the **Cycle 1 - New Features** cycle (created in Lesson 2, "Specify releases and cycles" on page 15).

   c. Click **OK**. The icon for the folder in the test sets tree changes to show that the folder has been assigned to a cycle.

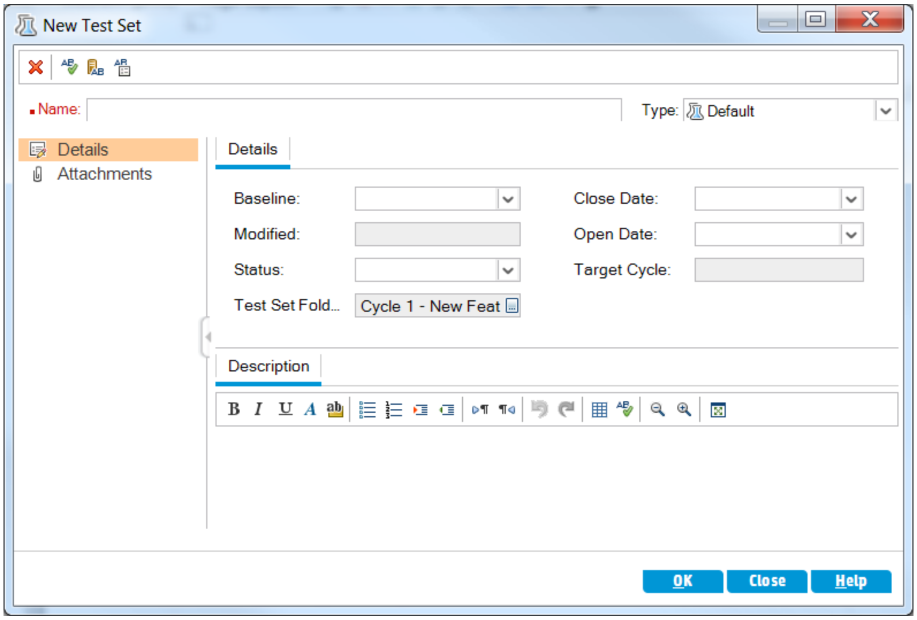


   d. Right-click the **Cycle 2 - Full** test sets folder and choose **Assign to Cycle**. Assign the folder to the **Cycle 2 - Full** cycle, located in the **Service Pack 1** release in the releases tree.

5. Add a test set to the Cycle 1 - New Features test set folder.

   a. In the test sets tree, select **Cycle 1- New Features**.

   b. Click the **New Test Set** button. The New Test Set dialog box opens.



Type: Default

Test Set Fold... Cycle 1 - New Feat

   c. Enter the following:

      Name: Mercury Tours Site

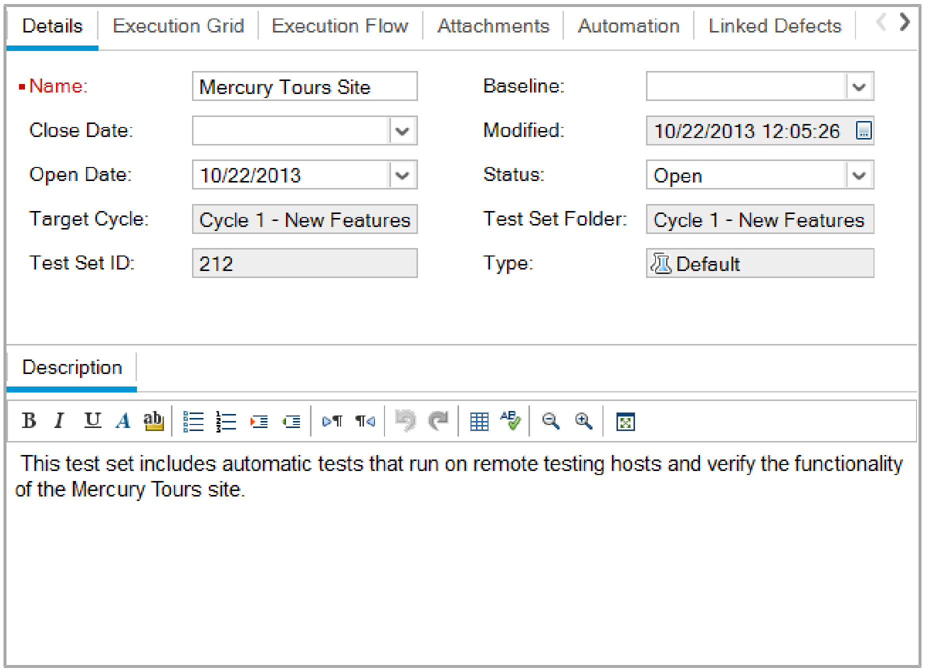
      Description: This test set includes automatic and manual tests that verify the functionality of the Mercury Tours site.

   d. Select **Default** in the **Type** field.

   e. Click **OK**. The **Mercury Tours Site** test set is added to the test sets tree in the left pane.

6. Define the Mercury Tours Site test set details.

   a. Click the test set in the test sets tree. The **Execution Grid** tab is displayed. Click on the **Details** tab.



Name:	Mercury Tours Site	Baseline:	
Close Date:		Modified:	10/22/2013 12:05:26
Open Date:	10/22/2013	Status:	Open
Target Cycle:	Cycle 1 - New Features	Test Set Folder:	Cycle 1 - New Features
Test Set ID:	212	Type:	Default

This test set includes automatic tests that run on remote testing hosts and verify the functionality of the Mercury Tours site.

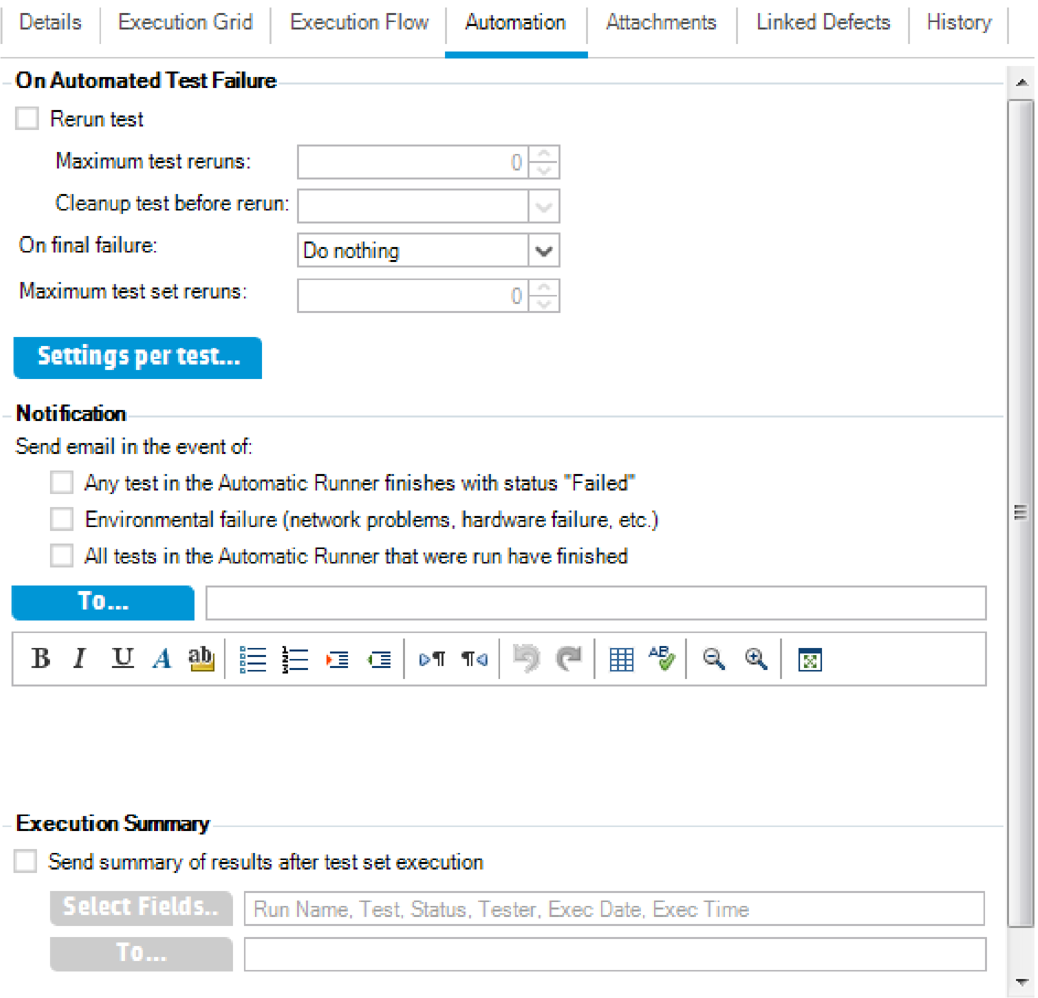
b. Select the following:

Open Date: Select a date from the calendar for the planned opening date for the test set. Today's date is selected by default.

Close Date: Select the planned closing date for the test set.

7. Set rules for the automated tests in the test set in the event of a test failure.

   a. Click the **Automation** tab.



On Automated Test Failure

□ Rerun test

Maximum test reruns: 0

Cleanup test before rerun:

On final failure: Do nothing

Maximum test set reruns: 0

Notification

Send email in the event of:

□ Any test in the Automatic Runner finishes with status "Failed"

□ Environmental failure (network problems, hardware failure, etc.)

□ All tests in the Automatic Runner that were run have finished

Execution Summary

□ Send summary of results after test set execution

Run Name, Test, Status, Tester, Exec Date, Exec Time

   b. Perform the following:

      - **On Automatic Test Failure**: Select the **Rerun test** check box. In **Maximum test reruns**, set to **1**.
      - **On final failure**: Make sure that the **Do nothing** option is selected.

8. Instruct ALM to send an email to specific users if certain events occur.

   Under **Notification**, perform the following:

   a. **Send email in the event of**: Select the first check box to send email notification if any test in the test set fails.

b. **To**: Enter your email address.

c. **Message**: Type the following:

   This test failed. Please review the test results and submit a defect.

Adding Tests to a Test Set

After you define a test set, select tests for inclusion in the test set. ALM adds instances of the selected tests to the test set. Each instance contains a defined test configuration. In this exercise, you will add tests to the **Mercury Tours Site** test set.

> **Note:**
>
> - If you are using ALM Edition with Lab Management enabled, add a test to your Functional test set. See "Adding Tests to a Functional Test Set" below.
> - If you are not using ALM Edition with Lab Management, add a test to your Default test set. See "Adding Tests to a Default Test Set" on the next page.

Adding Tests to a Functional Test Set

In this exercise, you add an automatic test to the **Mercury Tours Site** test set.

To add automatic tests to a Functional test set:

1. Display the **Execution Grid** tab.

   a. If the Test Lab module is not displayed, on the ALM sidebar, under **Testing**, select **Test Lab**.

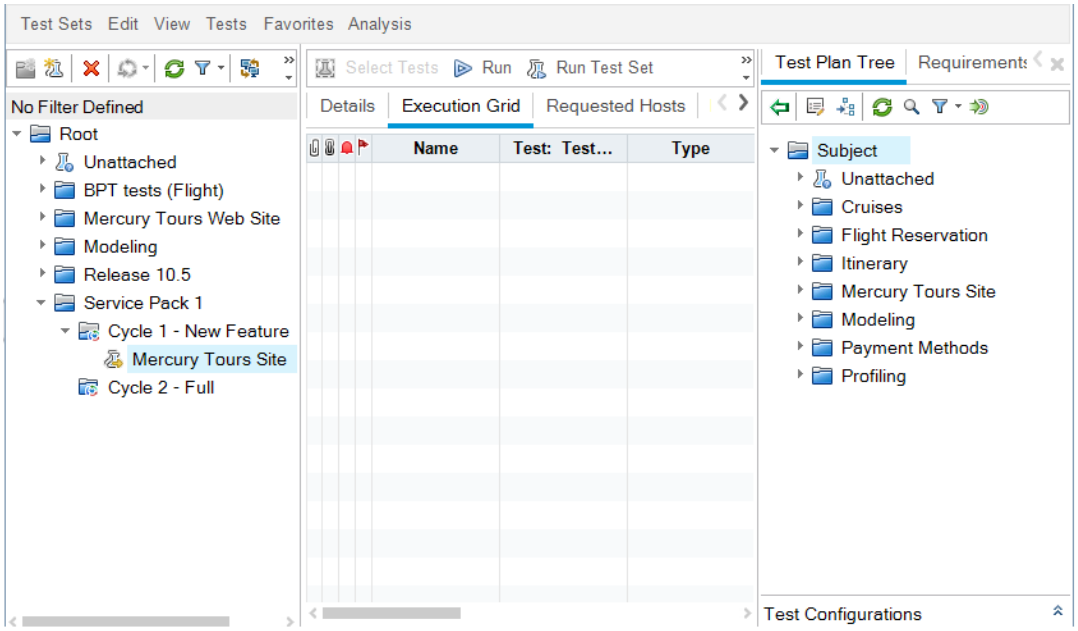
   b. Click the **Execution Grid** tab if it is not yet displayed.

2. Select the Mercury Tours Site test set.

   In the test sets tree, expand the **Cycle 1 - New Features** test set folder under **Service Pack 1**. Select the **Mercury Tours Site** test set.

3. Display the right pane if it is not already displayed.

Click the **Select Tests** button. The right pane displays the Test Plan Tree and Requirements Tree tabs.



The **Test Plan Tree** tab enables you to select tests from the test plan tree to add to the test set. The **Requirements Tree** tab enables you to select tests covering requirements to add to the test set.

4. Add the **Number of Passengers** test to the test set.

   a. Under the **Flight Reservation** folder, expand the **Flight Finder** folder.

   b. Drag the **Number of Passengers** test from the test plan tree to the Execution Grid to add it to the test set.

5. Close the right pane.

   Click the **Close** button.

Adding Tests to a Default Test Set

> **Note:** In this exercise, you add tests to a Default test set. If you are using ALM Edition with Lab Management, you already added tests to a Functional test set. Proceed to "Defining a Build Verification Suite" on page 68.

To add manual tests to a Default test set:

1. Display the **Execution Grid** tab.

   a. If the Test Lab module is not displayed, on the ALM sidebar, under **Testing**, select **Test Lab**.

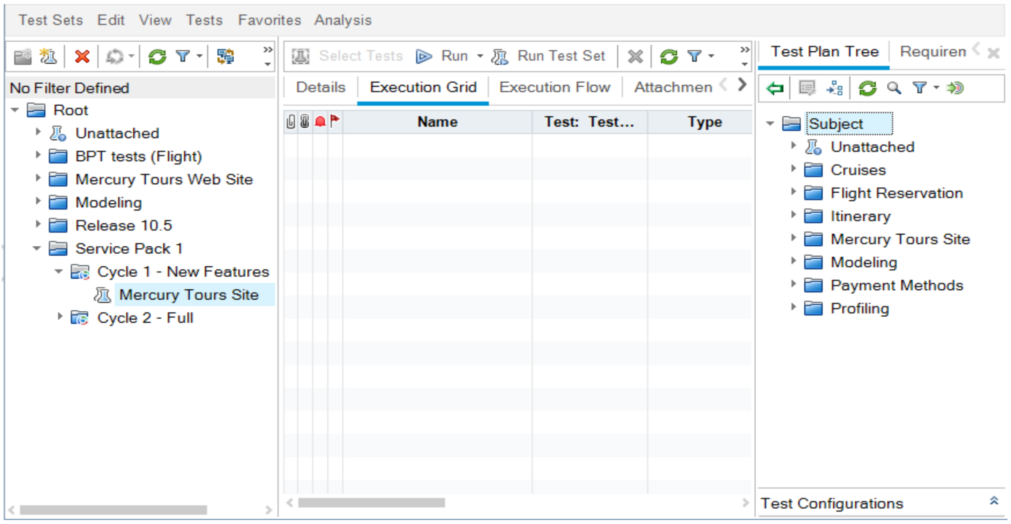
   b. Click the **Execution Grid** tab if it is not yet displayed.

2. Select the Mercury Tours Site test set.

   In the test sets tree, expand the **Cycle 1 - New Features** test set folder under **Service Pack 1**. Select the **Mercury Tours Site** test set.

3. Display the right pane if it is not already displayed.

   Click the **Select Tests** button. The right pane displays the Test Plan Tree and Requirements Tree tabs.

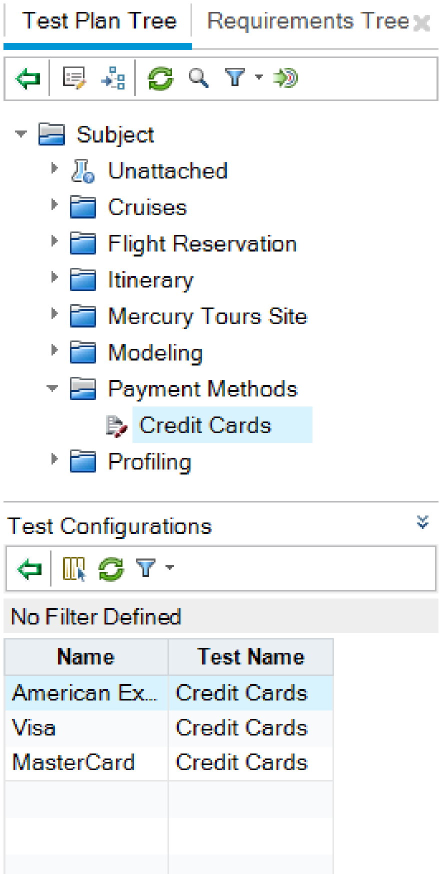


   The **Test Plan Tree** tab enables you to select tests from the test plan tree to add to the test set. The **Requirements Tree** tab enables you to select tests covering requirements to add to the test set.

4. Add the **Credit Card** test to the test set.

   a. In the **Test Plan Tree** tab, expand the **Payment Methods** folder and select the **Credit Cards** test.

b. If the Test Configurations pane is not displayed, click the **Show** button on the bottom of the pane. Under the Test Configurations pane, you can see the three test configurations for the selected test.



Name	Test Name
American Ex...	Credit Cards
Visa	Credit Cards
MasterCard	Credit Cards

c. To include all test configurations, in the Test Plan Tree tab, click the **Add Tests to Test Set** button. The instances are added to the test set.

5. Add several tests from the Book Flight folder to the test set.

   a. Under the **Flight Reservation** folder, expand the **Book Flight** folder.

   b. Select the **Passenger Name** test.

   c. Press the CTRL key and select the following tests: **Credit Card Number**, **Credit Card Expiration Date**, **Credit Card Owner**, and **Billing And Delivery Address**. Click the **Add Tests to Test Set** button. The Parameter Values dialog box opens.

   d. Click **Close**. The instances are added to the test set.

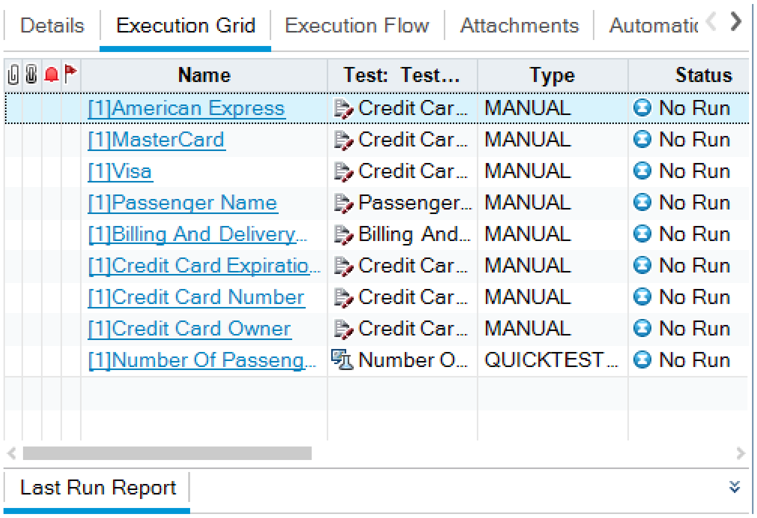
6. Add the Number of Passengers test to the test set.

   a. Under the **Flight Reservation** folder, expand the **Flight Finder** folder.

   b. Drag the **Number of Passengers** test from the test plan tree to the Execution Grid to add it to the test set.

7. Close the right pane.

   Click the **Close** button.



Name	Test: Test...	Type	Status
[1]American Express	Credit Car...	MANUAL	No Run
[1]MasterCard	Credit Car...	MANUAL	No Run
[1]Visa	Credit Car...	MANUAL	No Run
[1]Passenger Name	Passenger...	MANUAL	No Run
[1]Billing And Delivery...	Billing And...	MANUAL	No Run
[1]Credit Card Expiratio...	Credit Car...	MANUAL	No Run
[1]Credit Card Number	Credit Car...	MANUAL	No Run
[1]Credit Card Owner	Credit Car...	MANUAL	No Run
[1]Number Of Passeng...	Number O...	QUICKTEST...	No Run

Defining a Build Verification Suite

> **Note:** The Build Verification module is available only if you are using ALM Edition with the Lab Management extension enabled. If you are not using ALM Edition with Lab Management, proceed to "Setting Schedules and Conditions for Test Runs" on page 71.

The Build Verification module enables you to define a group of Functional test sets bundled together with a single Performance test. This group of test sets is called a build verification suite. When run together, the build verification suite checks the overall status of your build.

You can create a small suite to run immediately after a build in the middle of the day, create a suite with a few Functional test sets to run once every hour, or create a large suite to run for several hours every night.

Build verification suites are a key component in OpenText's Continuous Delivery solution. They facilitate an automated, end-to-end deployment and testing framework that makes application development more efficient, reliable, and quick.

In this exercise, you will create a build verification suite that includes Functional test sets.

To define a build verification suite:

1. Display the **Build Verification** module.

   On the ALM sidebar, under **Testing**, select **Build Verification**.

2. Add a folder to the Build Verification Suites tree.

   a. In the Build Verification Suites tree in the left pane, select the root **Build Verification Suites** folder.

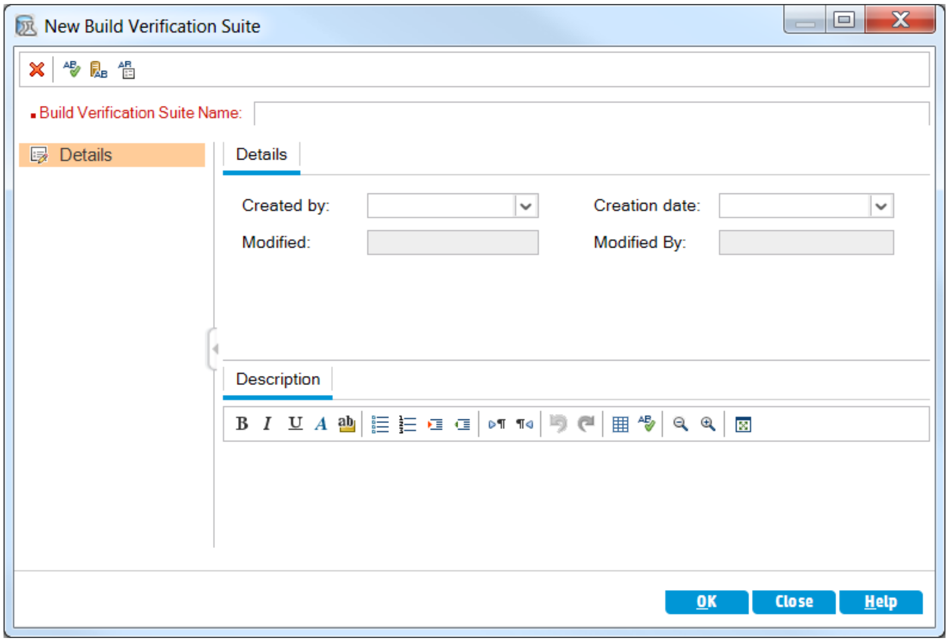
   b. Click the **New Folder** button. The New Build Verification Suite Folder dialog box opens.

   c. In the **Folder Name** box, type Mercury Tours Build Verification and click **OK**.

3. Add a build verification suite to the Mercury Tours Build Verification set folder.

   a. In the build verification suites tree, select **Mercury Tours Build Verification**.

   b. Click the **New Build Verification Suite** button. The New Build Verification Suite dialog box opens.



c. Type the following:

   Name: Mercury Tours Verification - Hourly

   Description: This build verification suite includes test sets that run on an hourly basis to verify the stability of the Mercury Tours site functionality.

d. Click **OK**. The **Mercury Tours Verification - Hourly** build verification suite is added to the test sets tree in the left pane.

4. Open the Functional Test Sets tab.

   In the build verification suites tree, select **Mercury Tours Verification - Hourly**. Select **Functional Test Sets** from the tabs in the right pane.

5. Add a Functional test set to the build verification suite.

   Click the **Select Test Sets** button. The right pane displays the Test Sets Tree tab. The Test Sets Tree tab enables you to select test sets from the test set tree to add to the build verification suite.

   Add the **Mercury Tours Site** test set to the build verification suite.

   a. Under the **Service Pack 1** folder, expand the **Cycle 1 - New Features** folder.

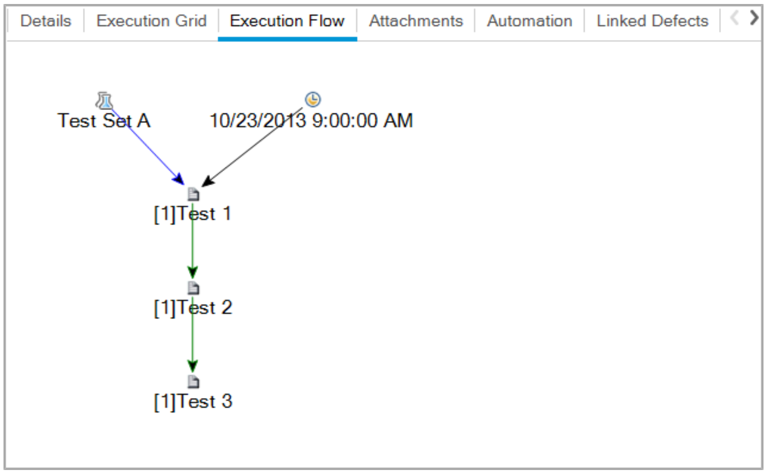
b. Drag the **Mercury Tours Site** test set from the test sets tree to the Functional Test Sets tab to add it to the build verification suite.

You can now schedule a timeslot to run this build verification suite.

Setting Schedules and Conditions for Test Runs

The Execution Flow tab enables you to specify a date and time to execute a test instance and set conditions for it. A **condition** is based on the results of another specified test instance in the Execution Flow. By setting conditions, you can postpone the execution of a test instance until another specified test instance finishes running or passes. You can also set the sequence in which to execute the test instances.

For example, you can determine that Test 2 will run only if Test 1 passed, and Test 3 will run only if Test 2 passed. Test 1 is scheduled to run at 9:00 AM on a specified date. The Execution Flow displays the tests and their conditions in a diagram.



A blue line arrow indicates that the test instance is to be executed after the previous test instance, with no conditions. A green line arrow indicates that the test instance is to be executed only if the previous test instance has status **Passed**. A black line arrow indicates that the test instance is to be executed only if the previous test instance has finished running. When a test instance is time-dependent, a Time Dependency 🕓 icon is added to the diagram.

In this exercise, you will create a new Default test set and add to it three test instances that verify the login procedure on the Sign-On page of the Mercury Tours site. Then, you will set the conditions for each instance and specify when each one is to be run.

To schedule a test run in the Execution Flow tab:

1. Make sure the Test Lab module is displayed.

   On the ALM sidebar, under **Testing**, select **Test Lab**.

2. Create a new test set.

   a. In the test sets tree, choose the **Service Pack 1** folder and click the **New Test Set** button. The New Test Set dialog box opens.

   b. Type the following:

      Name: Test Run Schedule

      Description: This test set is used to explain how to schedule a test run.

   c. Click **OK**. The **Test Run Schedule** test set is added to the test sets tree in the left pane.

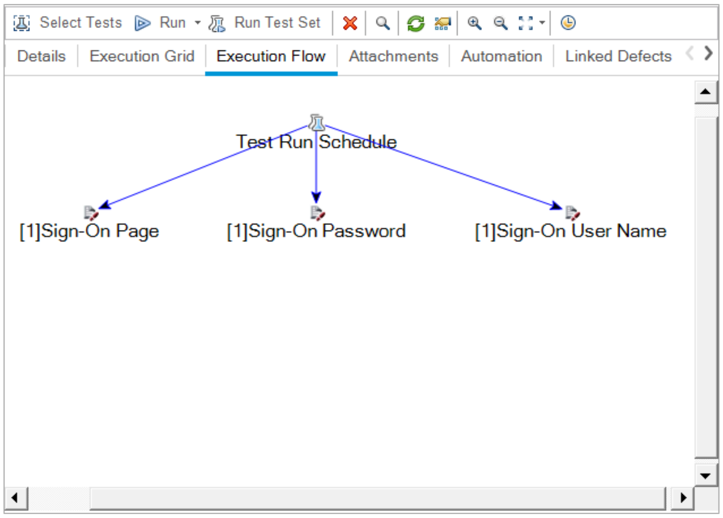
3. Add tests from the Sign-On/Sign-Off folder to the Test Run Schedule test set.

   a. Click the **Execution Flow** tab. If the right pane is not already displayed, click the **Select Tests** button. The Test Plan Tree tab and the Requirements Tree tab are displayed.

   b. In the Test Plan Tree tab, under the **Profiling** folder, expand the **Sign-On/Sign-Off** folder.

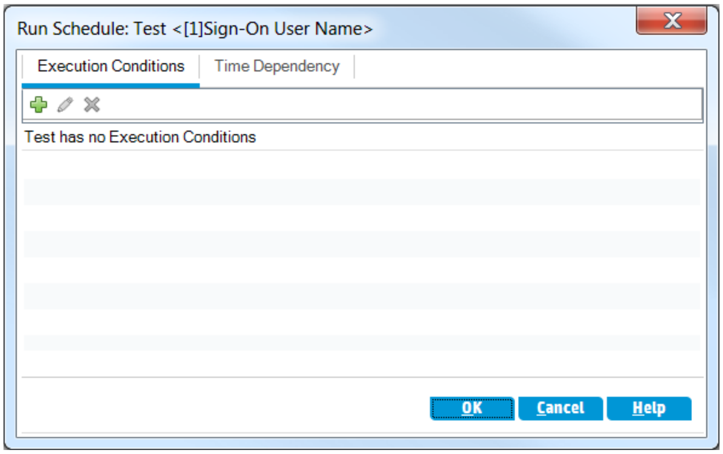
   c. Press the CTRL key and select the following tests: **Sign-On Page**, **Sign-On User Name**, and **Sign-On Password**. Click the **Add Tests to Test Set** button. The Parameter Values dialog box opens.

d. Click **Close**. The test instances are added to the test set.



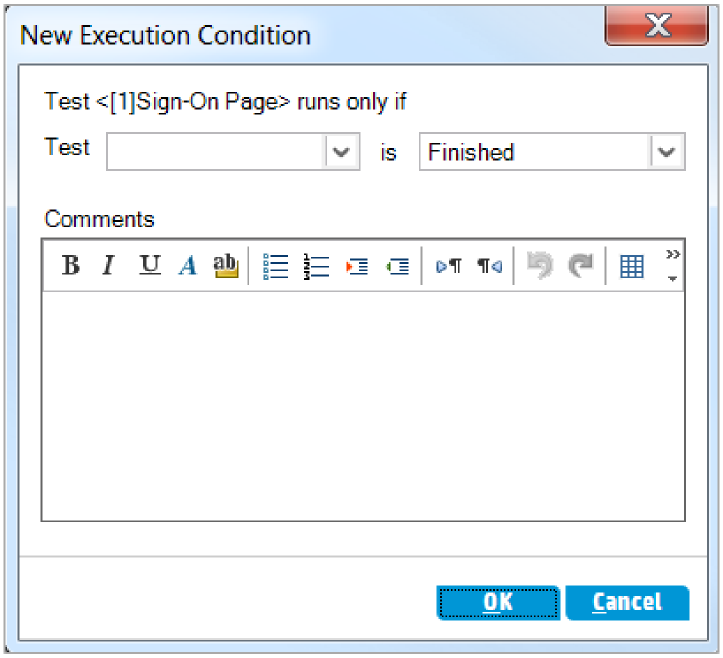
4. Add an execution condition to the Sign-On User Name test.

a. In the Execution Flow tab diagram, right-click the **Sign-On User Name** test instance and choose **Test Run Schedule**. The Run Schedule dialog box opens and displays the Execution Conditions tab.



Test has no Execution Conditions

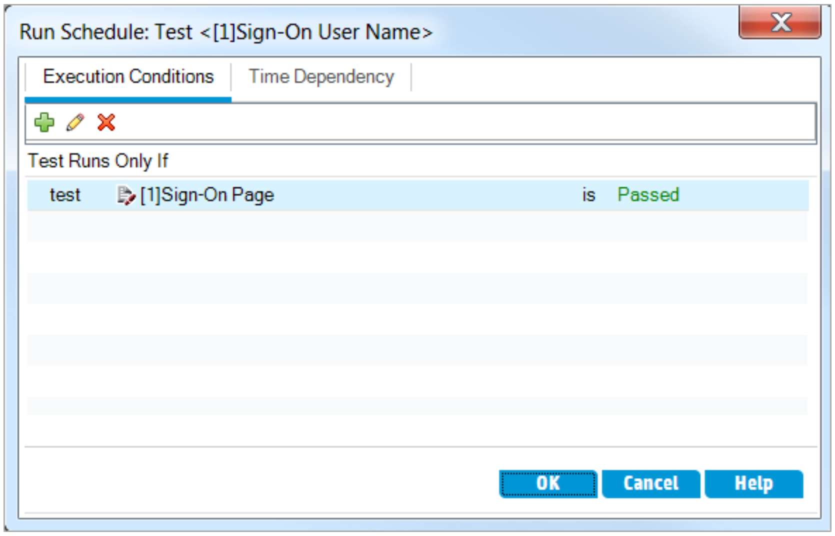
b. Click **New Execution Condition**. The New Execution Condition dialog box opens.



c. In the **Test** dropdown list, select **[1]Sign-On Page**.

d. Select **Passed** from the list on the right to instruct ALM to execute the **Sign-On User Name** test instance only if the **Sign-On Page** test instance finishes executing and passes.

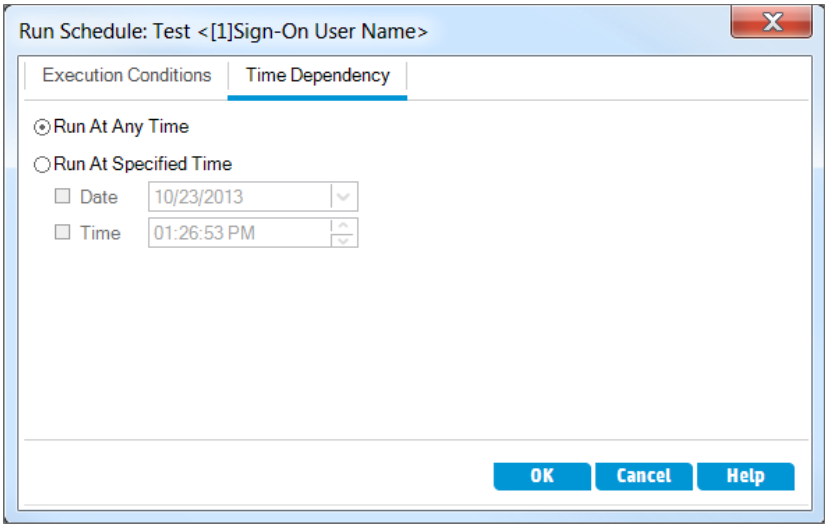
e. Click **OK**. The condition is added to the Run Schedule dialog box.



test [1]Sign-On Page is Passed

5. Add a time dependency condition to the Sign-On User Name test instance.

   a. Click the **Time Dependency** tab.



Run Schedule: Test <[1]Sign-On User Name>

● Run At Any Time

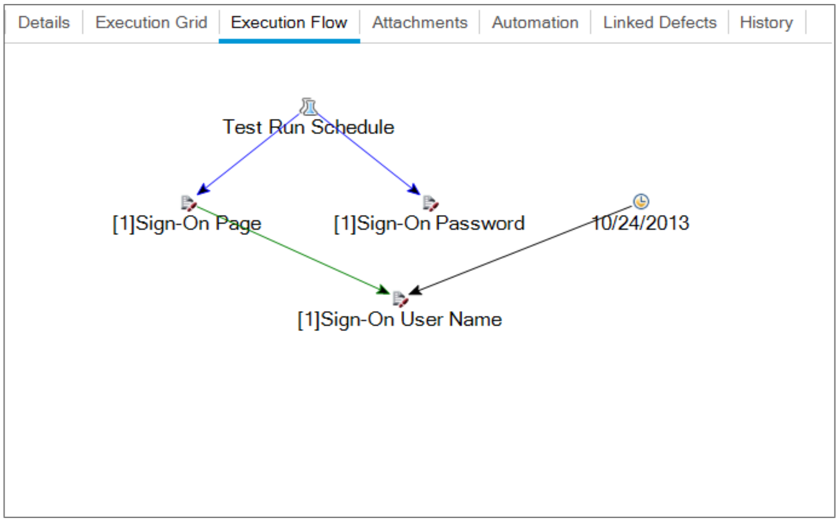
○ Run At Specified Time

□ Date 10/23/2013

□ Time 01:26:53 PM

   b. Click **Run At Specified Time**. Select the **Date** check box and select tomorrow’s date.

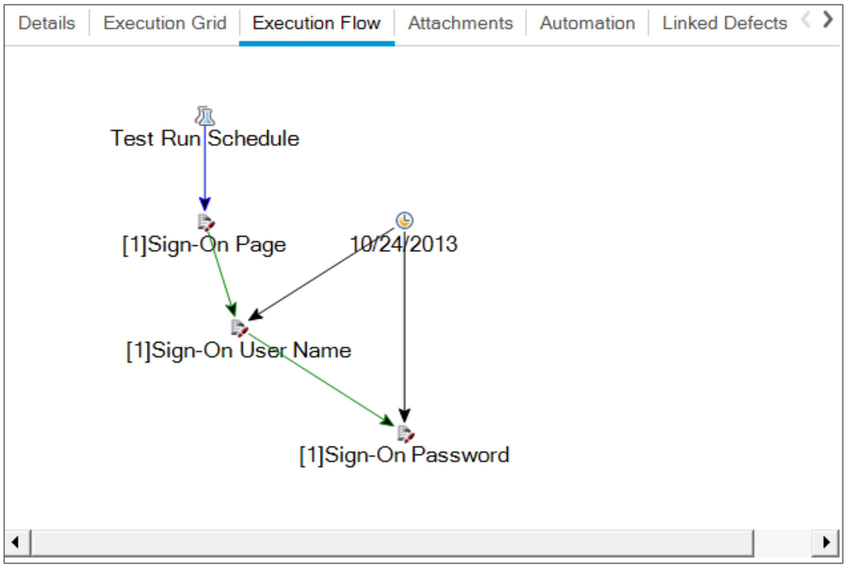
   c. Click **OK** to close the Run Schedule dialog box. Your conditions are displayed in the Execution Flow diagram.



6. Add an execution condition to the Sign-On Password test.

   Add the same execution condition as described in Step 4 for the **Sign-On Password** test. This time select **Sign-On User Name** from the **Test** box in the New Execution Condition dialog box.

7. Add a time dependency condition to the Sign-On Password test.
   a. Add the same time dependency condition as described in Step 5 for the **Sign-On Password** test.
   b. Click **OK** to close the Run Schedule dialog box. Your conditions are displayed in the Execution flow diagram.



Running Tests

In this exercise, you will define the Mercury Tours Site test set. You will also set failure rules for the test set to instruct ALM how to proceed in the event that an automated test in the test set fails. Depending on whether you are an ALM Edition user, you can either define a Functional test set or a Default test set.

Note:

- If you are using ALM Edition with the Lab Management extension enabled, run a test in your Functional test set. See "Running Tests in a Functional Test Set" on the next page
- If you are not using ALM Edition with Lab Management, run a test in your Default test set. See "Running Tests in a Default Test Set

Manually" on page 80 and "Running Tests in a Default Test Set Automatically" on page 86.

Running Tests in a Functional Test Set

When you run tests from a Functional test set, ALM uses Lab Management to execute the tests on remote testing hosts equipped with the testing tools. Lab Management updates ALM with the results of your tests. You can run all tests in a Functional test set or run specific tests. You can use the Execution Grid tab to run a test set immediately. You can also use the Timeslots module to reserve resources for a test to run in the future.

You can run build verification suites in the same way as test sets. You use the Build Verification module to run a build verification suite immediately, and you use the Timeslots module to reserve resources for a build verification suite to run in the future.

In these exercises, you will learn about the following:

- "Running a Functional Test Set in the Test Lab module" below
- "Scheduling a Functional Test Set in the Timeslots Module" on the next page

Running a Functional Test Set in the Test Lab module

You can run tests from a Functional test set immediately using the Execution Grid.

In this exercise, you will run the **Mercury Tours Site** test set in the Test Lab module.

To run a Functional test set in the Test Lab module:

1. Make sure the Test Lab module is displayed.

   On the ALM sidebar, under **Testing**, select **Test Lab**.

2. Select the Mercury Tours Site test set.

In the test set tree, expand the **Cycle 1 - New Features** test set folder under **Service Pack 1**. Select the **Mercury Tours Site** test set.

3. Open the Run Functional Test Set dialog box.

   Click the **Run Test Set** button. The Run Functional Test Set dialog box opens.

4. Run the test set.

   Click the **Run** button. ALM uses Lab Management to execute your test on a testing host with the required testing tool.

5. View the progress of the run.

   ALM opens the Execution Report page, which displays the current state and results of the tests you executed. You can refresh the page, stop tests, and view an Event Log for the entire run.

6. Close the Execution Report.

   After the test run is complete, close the Execution Report page.

Scheduling a Functional Test Set in the Timeslots Module

You can reserve resources for the execution of a Functional test set using the Timeslots module.

In this exercise, you schedule the **Mercury Tours Site** test set in the Timeslots module.

To schedule a Functional test set in the Timeslots module:

1. Make sure the Timeslots module is displayed.

   On the ALM sidebar, under **Testing**, select **Timeslots**.

2. Create a new Timeslot.

   On the Timeslots toolbar, click the **New Timeslot** button. The Timeslot Reservation dialog box opens.

3. Schedule your Functional test set.

a. In the **Run** field, select **Functional Test Set**.

b. In the **Start** field, select **Automatically**.

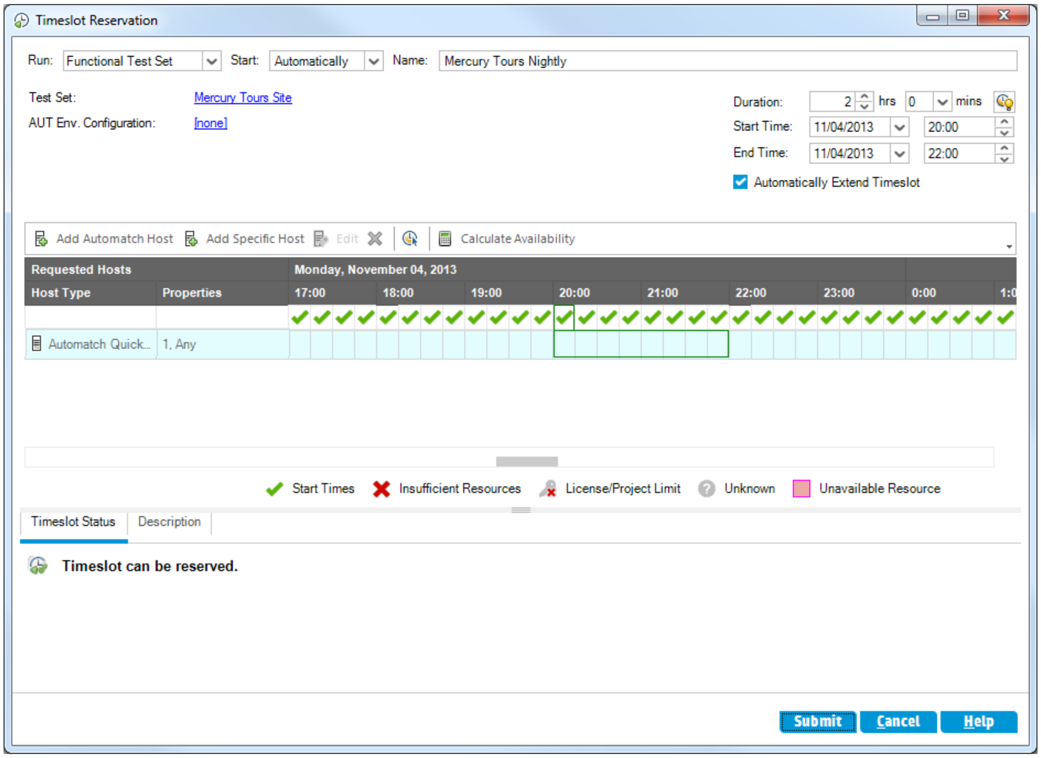
c. In the **Name** field, type Mercury Tours Nightly.

d. Click the **[none]** link in **Select a test set**, and select the **Mercury Tours Site** test set.

e. Use the **Start Time** and **End Time** fields to schedule the test set to run from 20:00 to 22:00.

f. A host is automatically added to the **Requested Hosts** grid.

g. Click the **Calculate Availability** button to check that the timeslot is valid.



Run: Functional Test Set Start: Automatically Name: Mercury Tours Nightly

Test Set: Mercury Tours Site

AUT Env. Configuration: [none]

Duration: 2 hrs 0 mins

Start Time: 11/04/2013 20:00

End Time: 11/04/2013 22:00

☑ Automatically Extend Timeslot

Requested Hosts	
Host Type	Properties
Automatch Quick...	1, Any

✓ Start Times ✗ Insufficient Resources License/Project Limit Unknown Unavailable Resource

Timeslot can be reserved.

4. Submit the timeslot.

Click **Submit**.

Running Tests in a Default Test Set Manually

When you run a test manually, you follow the test steps and perform operations on your application. Then, you compare the expected results with the actual outcome and record the results. You can execute a manual test as many times as needed. The results are stored separately for each run.

> **Note:** In these exercises, you run Default test sets. If you are using ALM Edition with Lab Management, you already ran a Functional test set. To continue in the tutorial, proceed to "Viewing and Analyzing Test Results" on page 88.

You can run both manual and automated tests manually as part of a Default test set. You can also choose to run a single test or to run an entire test set.

You run manual tests in ALM using Sprinter, OpenText's solution for manual testing. If you are not working with Sprinter, you run tests manually using Manual Runner.

In these exercises, you will learn about the following:

- "Running with Sprinter" below
- "Running with Manual Runner" on page 83

Running with Sprinter

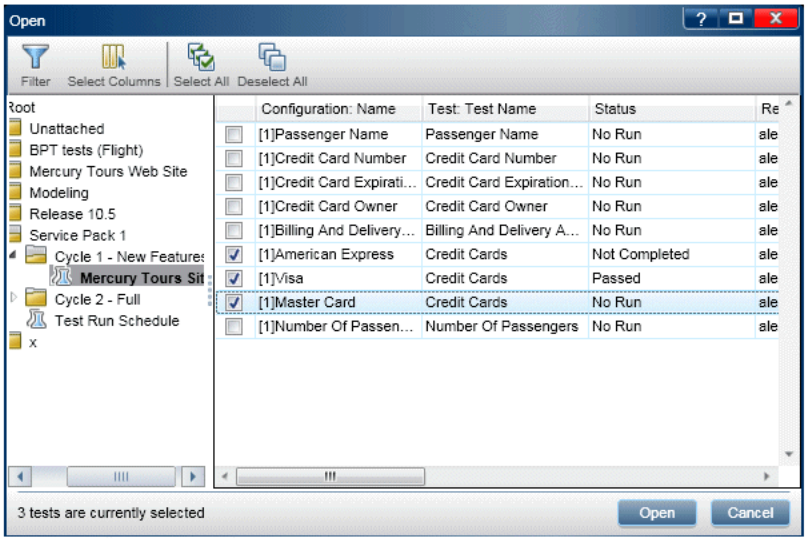
Sprinter provides advanced functionality and tools to assist you in the manual testing process. Sprinter is fully integrated with ALM, enabling you to get the maximum benefit from both solutions.

> **Note:** For information on installing Sprinter, see "Tutorial " on page 6.

In this exercise, you will run the **Credit Cards** test. This test contains three test configurations. For the purpose of this exercise, you will perform the steps without testing them against the Mercury Tours application.

To run a test using Sprinter:

1. Open Sprinter.
   a. If the Test Lab module is not displayed, on the ALM sidebar, under **Testing**, select **Test Lab**.
   b. Click the **Execution Grid** tab.
   c. Click the **Run** arrow and select **Run with Sprinter**. OpenText Sprinter opens.
2. Select the instances to run from the test set.
   a. Click the **Open ALM Tests** button. The Open dialog box opens.
   b. In the left pane, expand the **Root** folder. Under **Service Pack 1**, expand **Cycle 1 - New Features**. Select the **Mercury Tours Site** test set. The test set is displayed.



Root
- Unattached
- BPT tests (Flight)
- Mercury Tours Web Site
- Modeling
- Release 10.5
- Service Pack 1
  - Cycle 1 - New Features
    - Mercury Tours Sit
  - Cycle 2 - Full
  - Test Run Schedule
- x

	Configuration: Name	Test: Test Name	Status	Re
☐	[1]Passenger Name	Passenger Name	No Run	ale
☐	[1]Credit Card Number	Credit Card Number	No Run	ale
☐	[1]Credit Card Expirati...	Credit Card Expiration...	No Run	ale
☐	[1]Credit Card Owner	Credit Card Owner	No Run	ale
☐	[1]Billing And Delivery...	Billing And Delivery A...	No Run	ale
☑	[1]American Express	Credit Cards	Not Completed	ale
☑	[1]Visa	Credit Cards	Passed	ale
☑	[1]Master Card	Credit Cards	No Run	ale
☐	[1]Number Of Passen...	Number Of Passengers	No Run	ale

3 tests are currently selected

   c. Select **American Express**, **Visa**, and **MasterCard** check boxes. Click **Open**.
3. Display the test steps to run the American Express instance.

   Click the **Run the Active Test** button. The Steps pane is displayed.
4. Perform the first step.
   a. Click the **Actual Result** button. In the Actual Result dialog box, type: The Mercury Tours site opens. Click **OK**.

   b. Click the **Passed Selected Step** button.

5. Perform the second step.

   a. Click the **Actual Result** button. In the Actual Result dialog box, type: Flight details and preference are entered. Click **OK**.

   b. Click the **Passed Selected Step** button.

6. Pass the remaining steps.

   Click the **Passed Selected Step** arrow and select **Pass All**.

7. Continue on to the Visa instance.

   Click **Next Test**. Sprinter advances to the next instance in the test list.

8. Pass all steps of the Visa instance.

   Click the **Passed Selected Step** arrow and select **Pass All**.

9. Continue on to the MasterCard instance.

   Click **Next Test**. Sprinter advances to the next instance in the test list.

10. Pass all steps of the MasterCard instance.

    Click the **Passed Selected Step** arrow and select **Pass All**.

11. Fail the last step on the MasterCard instance.

    Select **Step 8** and click the **Fail Selected Step** button.

12. End the run and view run results.

    a. In the upper-right side of your screen, click **Run Control**. The Run Control pane opens.

    b. Click the **End Run** button.

    c. Under **Tests**, you view the run results.

13. View results in the Execution Grid tab.

    Close **Sprinter**.

    The test run results are displayed in the execution grid. The Last Run Report pane displays run results of each test step.

Running with Manual Runner

If Sprinter is not installed you can run tests manually using Manual Runner.

In this exercise, you will run the **Credit Cards** test. This test contains three test configurations. For the purpose of this exercise, you will perform the steps without testing them against the Mercury Tours application.

To run a test using Manual Runner:

1. Make sure the **Cycle 1 - New Features** test set folder is displayed in the Execution Grid.

   a. If the Test Lab module is not displayed, on the ALM sidebar, under **Testing**, select **Test Lab**.

   b. In the test set tree, expand the **Cycle 1 - New Features** test set folder under **Service Pack 1**. Select the **Mercury Tours Site** test set.

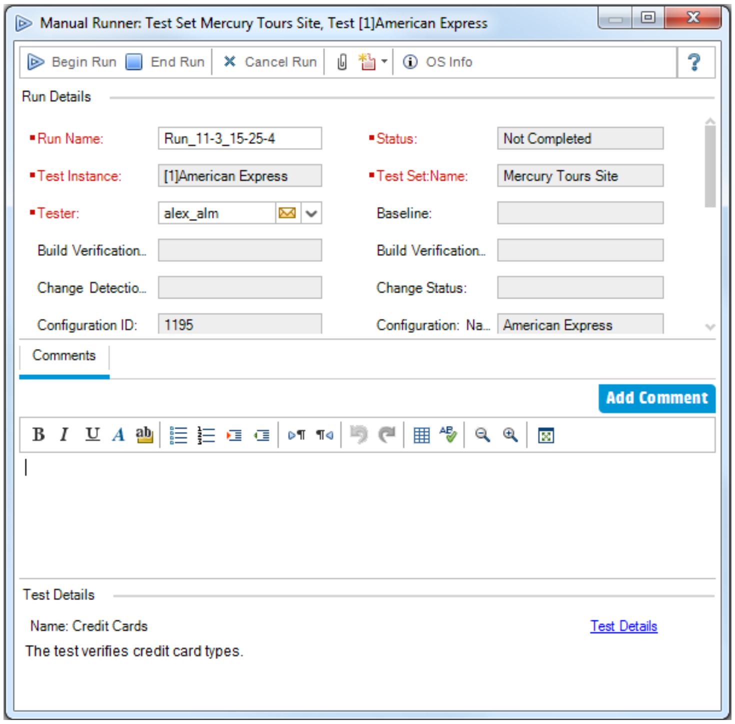
   c. Click the **Execution Grid** tab.

2. Select the instances to run from the test set.

   Press the CTRL key and select the following instances in the Execution Grid: **American Express**, **Visa**, and **MasterCard**.

3. Open Manual Runner.

Click the **Run** arrow and select **Run with Manual Runner**. The Manual Runner dialog box opens.



▪Run Name:	Run_11-3_15-25-4	▪Status:	Not Completed
▪Test Instance:	[1]American Express	▪Test Set:Name:	Mercury Tours Site
▪Tester:	alex_alm	Baseline:	
Build Verification..		Build Verification..	
Change Detectio..		Change Status:	
Configuration ID:	1195	Configuration: Na..	American Express

Name: Credit Cards

The test verifies credit card types.

4. Start the test run.

   Click the **Begin Run** button. The Manual Runner dialog box opens.

5. Perform the first step.

   a. In the **Actual** box, type: The Mercury Tours site opens.

   b. Click the **Pass Selected** button. Step 2 is displayed.

6. Perform the second step.

   a. In the **Actual** box, type: Flight details and preference are entered.

   b. Click the **Pass Selected** button. Step 3 is displayed.

7. Pass the remaining steps.

   Click the **Passed Selected** arrow and select **Pass All**.

8. End the run.

   Click the **End Run** button to end your test run.

9. Continue on to the Visa instance.

Click the **Begin Run** button. The Manual Runner dialog box opens. Note the name of the instance in the title bar.

10. Pass all steps of the Visa instance.

    Click the **Passed Selected** arrow and select **Pass All**.

11. End the run.

    Click the **End Run** button to end your test run.

12. Continue on to the MasterCard instance.

    Click the **Begin Run** button. The Manual Runner dialog box opens. Note the name of the instance in the title bar.

13. Fail all steps of the MasterCard instance.

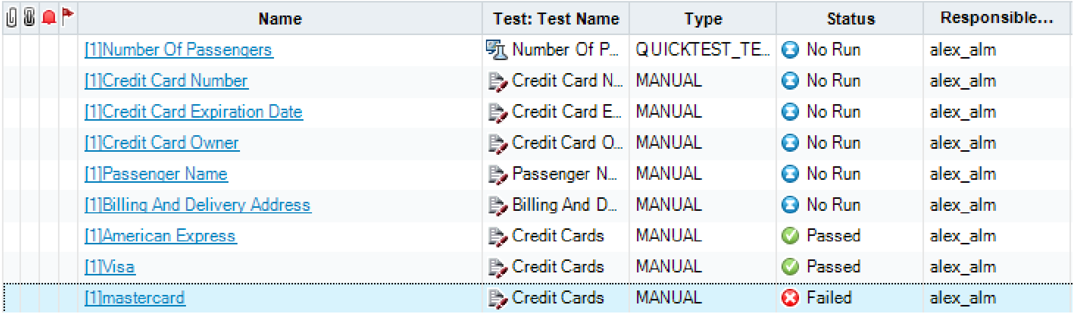
    Click the **Fail Selected** arrow and select **Fail All**.

14. End the run.

    Click the **End Run** button to end your test run.

15. View the run results in the Execution Grid.

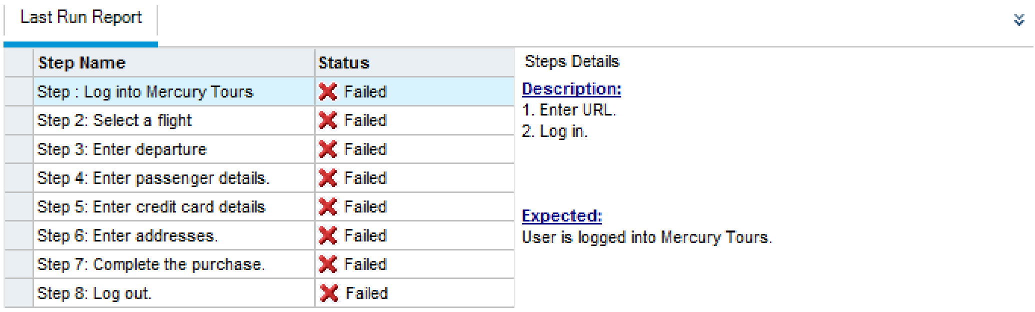
    Following the execution of your tests, you can view the run results of your last run in the Execution Grid.



Name	Test: Test Name	Type	Status	Responsible...
[1]Number Of Passengers	Number Of P...	QUICKTEST_TE...	No Run	alex_alm
[1]Credit Card Number	Credit Card N...	MANUAL	No Run	alex_alm
[1]Credit Card Expiration Date	Credit Card E...	MANUAL	No Run	alex_alm
[1]Credit Card Owner	Credit Card O...	MANUAL	No Run	alex_alm
[1]Passenger Name	Passenger N...	MANUAL	No Run	alex_alm
[1]Billing And Delivery Address	Billing And D...	MANUAL	No Run	alex_alm
[1]American Express	Credit Cards	MANUAL	Passed	alex_alm
[1]Visa	Credit Cards	MANUAL	Passed	alex_alm
[1]mastercard	Credit Cards	MANUAL	Failed	alex_alm

16. View the results of each test step in the Last Run Report pane.

    a. Select one of the recently run instances. If the Last Run Report pane is not displayed, click the **Show** button on the bottom of the pane. The Last Run Report pane is displayed below the Execution Grid.



Step Name	Status
Step : Log into Mercury Tours	✗ Failed
Step 2: Select a flight	✗ Failed
Step 3: Enter departure	✗ Failed
Step 4: Enter passenger details.	✗ Failed
Step 5: Enter credit card details	✗ Failed
Step 6: Enter addresses.	✗ Failed
Step 7: Complete the purchase.	✗ Failed
Step 8: Log out.	✗ Failed

Steps Details

Description:
1. Enter URL.
2. Log in.

Expected:
User is logged into Mercury Tours.

b. Click each step to view its description, as well as the expected and actual results.

Running Tests in a Default Test Set Automatically

When you run an automated test from a Default test set, ALM automatically opens the selected testing tool, which runs the test on your local machine or on remote hosts, and imports the results to ALM.

You can run all tests in a test set or run specific tests. You can run tests from the Execution Grid tab or the Execution Flow tab.

In this exercise, you will run a UFT test.

> **Note:** For prerequisites to running a UFT test, see "Tutorial " on page 6.

To run a test automatically:

1. Make sure the Test Lab module is displayed.

   On the ALM sidebar, under **Testing**, select **Test Lab**.

2. Select the **Number of Passengers** test.

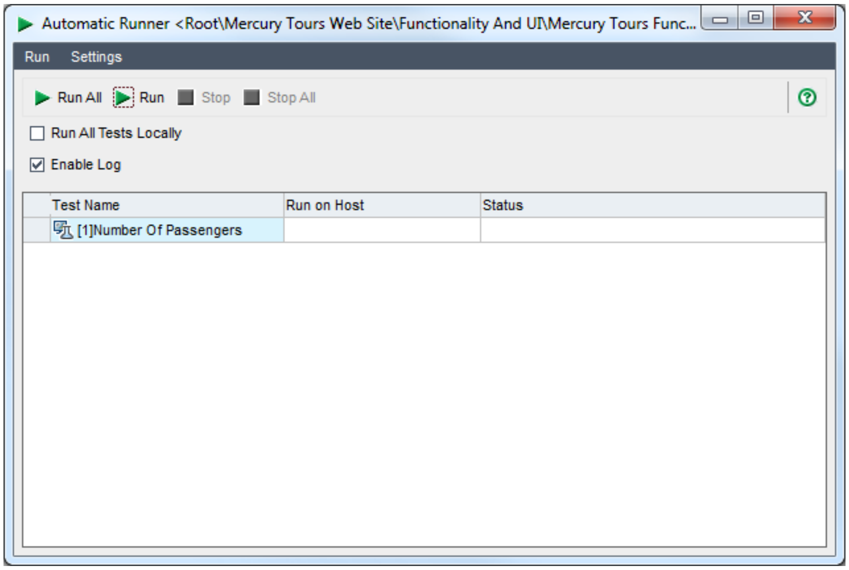
   a. In the test sets tree, expand the **Mercury Tours Web Site** test set folder. Under **Functionality And UI**, select the **Mercury Tours Functionality** test set.

   b. Click the **Execution Grid** tab.

   c. Select the **Number of Passengers** test.

3. Open the Automatic Runner dialog box.

   Click the **Run** button. The Automatic Runner dialog box opens and displays the selected test.


Automatic Runner <Root\Mercury Tours Web Site\Functionality And UI\Mercury Tours Func...
Run
Settings
Run All
Run
Stop
Stop All

□ Run All Tests Locally

☑ Enable Log

Test Name	Run on Host	Status
[1]Number Of Passengers		

4. Set the test run settings.

   Select the **Run All Tests Locally** check box to run the test on your local computer.

5. Run the test.

   Click the **Run** button. ALM opens the selected testing tool automatically and runs the test. You view the test execution progress in the **Status** column.

6. Close the Automatic Runner dialog box.

   After the test run is complete, choose **Run > Exit**.

7. View a summary of test results in the Execution Grid.

   The Execution Grid displays the updated status for the test run. Results for each test step appear in the Last Run Report pane.

8. Close UFT.

   In UFT, choose **File > Exit**.

Viewing and Analyzing Test Results

ALM provides a number of features that enable you to view and analyze the results of your tests.

This section includes:

- "Viewing Test Results in the Test Runs Module" below
- "Viewing Test Results in the Test Instance Properties Dialog Box" on page 91
- "Viewing Test Coverage" on page 92
- "Viewing Coverage Progress" on page 95

You can also use ALM reports and graphs to further analyze your test results. For details, see "Analyzing Data" on page 119.

Viewing Test Results in the Test Runs Module

You can view results for a test in the Test Runs module. You can use the grid to compare the results of recent test runs with previous test runs.

In the below exercise, you will learn how to view test run information in the Test Runs module.

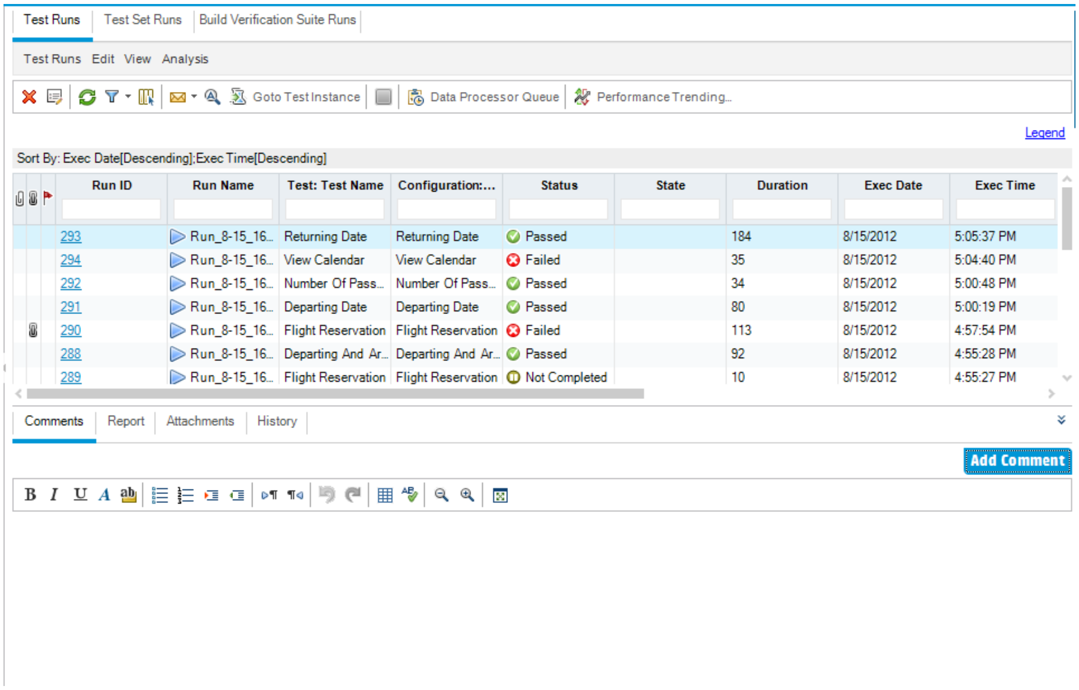
Viewing Functional Test Set Results in the Test Set Runs Tab

If you are using ALM Edition with Lab Management enabled, you can view the results of your Functional test set runs in the Test Set Runs tab.

To view test set results in the Test Set Runs tab:

1. Make sure the Test Runs module is displayed.

   On the ALM sidebar, under **Testing**, select **Test Runs**.



Sort By: Exec Date[Descending];Exec Time[Descending]

	Run ID	Run Name	Test: Test Name	Configuration:...	Status	State	Duration	Exec Date	Exec Time
	293	Run_8-15_16...	Returning Date	Returning Date	✓ Passed		184	8/15/2012	5:05:37 PM
	294	Run_8-15_16...	View Calendar	View Calendar	☒ Failed		35	8/15/2012	5:04:40 PM
	292	Run_8-15_16...	Number Of Pass...	Number Of Pass...	✓ Passed		34	8/15/2012	5:00:48 PM
	291	Run_8-15_16...	Departing Date	Departing Date	✓ Passed		80	8/15/2012	5:00:19 PM
	290	Run_8-15_16...	Flight Reservation	Flight Reservation	☒ Failed		113	8/15/2012	4:57:54 PM
	288	Run_8-15_16...	Departing And Ar...	Departing And Ar...	✓ Passed		92	8/15/2012	4:55:28 PM
	289	Run_8-15_16...	Flight Reservation	Flight Reservation	○ Not Completed		10	8/15/2012	4:55:27 PM

2. View detailed test results from the Test Set Run Details dialog box.

   Click the **Test Set Runs** tab. Select the **Mercury Tours Site** test set run in the grid and click the **Test Set Run Details** button. The Test Set Run Details dialog box opens, containing details about the test set run.

3. Close the Test Set Run Details dialog box.

   Click the **Close** button.

4. View test run information in the Execution Report page.

   Select the **Mercury Tours Site** test run and click the **Show Report** button. The **Execution Report** page opens. The Execution Report page displays information about the overall test set run and detailed information about each test instance run.

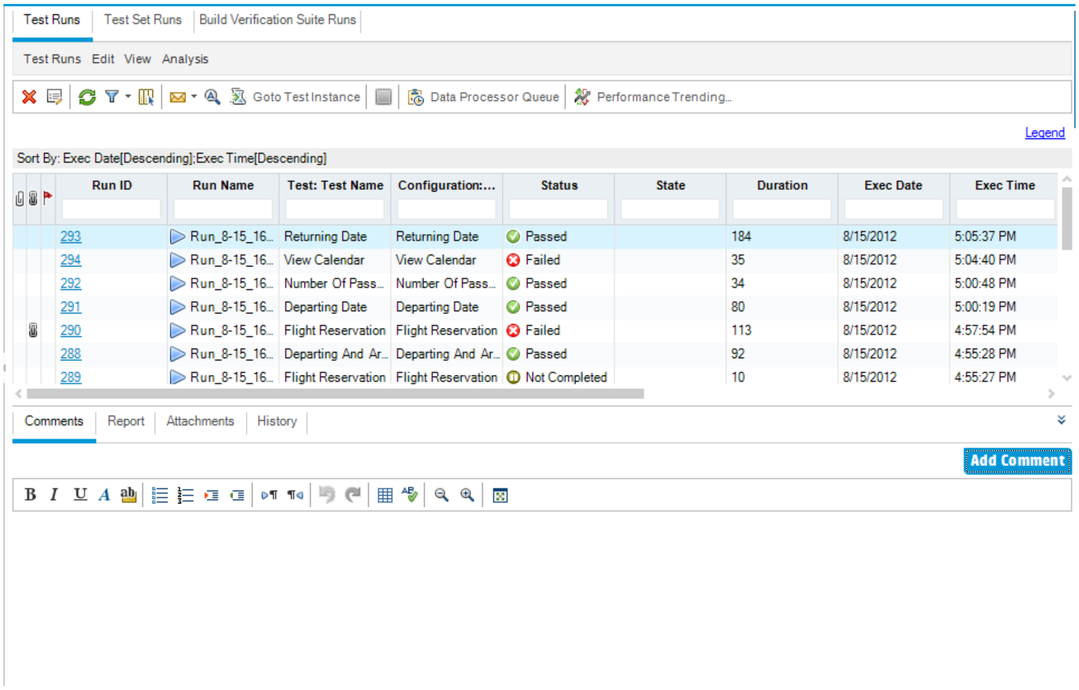
Viewing Test Results in the Test Runs Tab

In this exercise, you view the results of individual test instance runs.

To view test results in the Test Runs tab:

1. Make sure the Test Runs module is displayed.

   On the ALM sidebar, under **Testing**, select **Test Runs**.

   Sort By: Exec Date[Descending];Exec Time[Descending]

Run ID	Run Name	Test: Test Name	Configuration:...	Status	State	Duration	Exec Date	Exec Time
293	Run_8-15_16...	Returning Date	Returning Date	✓ Passed		184	8/15/2012	5:05:37 PM
294	Run_8-15_16...	View Calendar	View Calendar	✗ Failed		35	8/15/2012	5:04:40 PM
292	Run_8-15_16...	Number Of Pass...	Number Of Pass...	✓ Passed		34	8/15/2012	5:00:48 PM
291	Run_8-15_16...	Departing Date	Departing Date	✓ Passed		80	8/15/2012	5:00:19 PM
290	Run_8-15_16...	Flight Reservation	Flight Reservation	✗ Failed		113	8/15/2012	4:57:54 PM
288	Run_8-15_16...	Departing And Ar...	Departing And Ar...	✓ Passed		92	8/15/2012	4:55:28 PM
289	Run_8-15_16...	Flight Reservation	Flight Reservation	Not Completed		10	8/15/2012	4:55:27 PM

2. View detailed test results from the Test Run Details dialog box.

   Make sure the **Test Runs** tab is displayed. Select the **American Express** test run and click the **Test Run Details** button. The Test Run Details dialog box opens.

3. View other test run information in the Test Run Details dialog box.

   a. Click **Report** to display results and test step details of the run.

   b. Click **Linked Defects** to list the defects linked to the run.

   c. Click **History** to list the changes made to the run.

4. Close the Test Set Run Details dialog box.

   Click the **Close** button.

Viewing Test Results in the Test Instance Properties Dialog Box

You can view results for a test in the Test Instance Properties dialog box. This includes details of a test’s runs, attachments, linked defects, and history.

To view test results in the Test Instance Properties dialog box:

1. Make sure the Test Lab module is displayed.

   On the ALM sidebar, under **Testing**, select **Test Lab**.

2. Make sure the **Cycle 1 - New Feature**s test set folder is displayed.

   a. In the test set tree, expand the **Cycle 1 - New Features** test set folder under **Service Pack 1**. Select the **Mercury Tours Site** test set.

   b. Click the **Execution Grid** tab.

3. Select American Express in the Execution Grid.

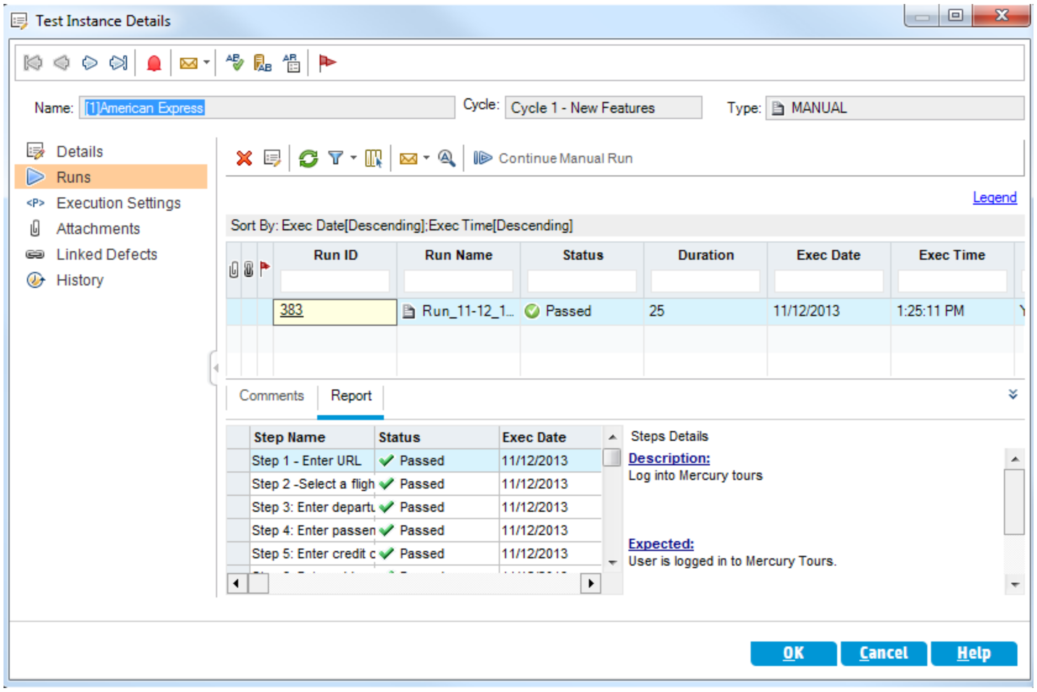
   In the Execution Grid tab, select **American Express**.

4. View detailed test results from the Test Instance Details dialog box.

   Click the **Test Instance Details** button. The Test Instance Details dialog box opens.

5. View test run information.

Click **Runs**. The run details are displayed.



Name: [1]American Express

Cycle: Cycle 1 - New Features

Type: MANUAL

Sort By: Exec Date[Descending];Exec Time[Descending]

Run ID	Run Name	Status	Duration	Exec Date	Exec Time
383	Run_11-12_1...	Passed	25	11/12/2013	1:25:11 PM

Step Name	Status	Exec Date
Step 1 - Enter URL	✓ Passed	11/12/2013
Step 2 -Select a fligh	✓ Passed	11/12/2013
Step 3: Enter departu	✓ Passed	11/12/2013
Step 4: Enter passen	✓ Passed	11/12/2013
Step 5: Enter credit c	✓ Passed	11/12/2013

Steps Details

Description:
Log into Mercury tours

Expected:
User is logged in to Mercury Tours.

6. View other test run information in the Test Instance Details dialog box.
   a. Click **Execution Settings** to view parameters used to run the instance. Note that any changes that you make are implemented in the next test run.
   b. Click **Linked Defects**. This view lists the defects linked to the currently selected test instance. You can view, add, and remove defect links.
   c. Click **History** to view a list of changes made to the test run fields.
7. Close the Test Instance Details dialog box.

   Click the **Close** button.

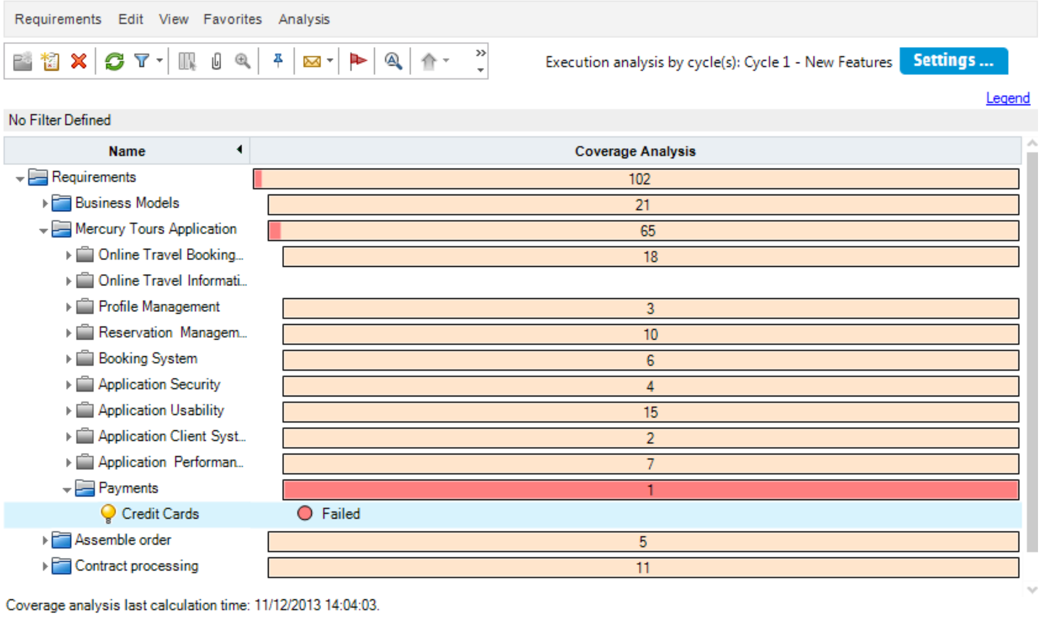
Viewing Test Coverage

You previously saw how you can use the Coverage Analysis view to analyze the breakdown of child requirements according to their tests coverage (see "Analyzing Coverage" on page 45).

In this exercise, you will learn how to filter the tests included in the coverage analysis by cycle.

To view test coverage:

1. Display the Coverage Analysis view in the Requirements module.
   a. On the ALM sidebar, under **Requirements**, select **Requirements**.
   b. Choose **View > Coverage Analysis**. The Coverage Analysis view is displayed.
2. Filter the coverage analysis by cycle.
   a. On the upper-right side of the window, click the **Settings** button. The Analysis Settings dialog box opens.
   b. Select **Execution Analysis**. Click the arrow. In the releases tree, expand the release folder **Service Packs** and the release **Service Pack 1**. Select the **Cycle 1 - New Features** check box and click **OK**.
   c. Click **OK** to close the Analysis Settings dialog box.
   d. Expand the **Payments** requirement. In the Coverage Analysis column, you can see that the requirement has failed. This is because the Credit Cards test that covers the requirement failed.



Execution analysis by cycle(s): Cycle 1 - New Features

No Filter Defined

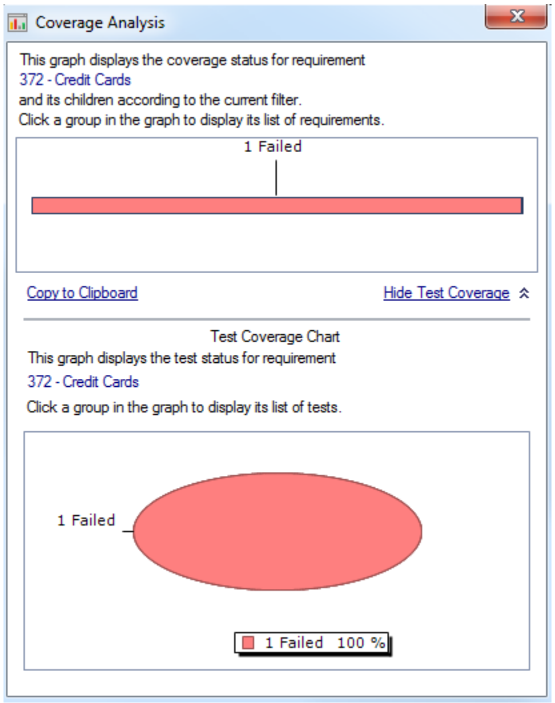
Name	Coverage Analysis
Requirements	102
Business Models	21
Mercury Tours Application	65
Online Travel Booking...	18
Online Travel Informati...	
Profile Management	3
Reservation Managem...	10
Booking System	6
Application Security	4
Application Usability	15
Application Client Syst...	2
Application Performan...	7
Payments	1
Credit Cards	● Failed
Assemble order	5
Contract processing	11

Coverage analysis last calculation time: 11/12/2013 14:04:03.

3. Display test coverage details for the Credit Cards requirement.

   a. Right-click the **Credit Cards** requirement, and choose **Coverage Analysis**. The Coverage Analysis dialog box opens.

   b. Click the **Show Test Coverage** link to extend the Coverage Analysis dialog box and display the Test Coverage Chart.

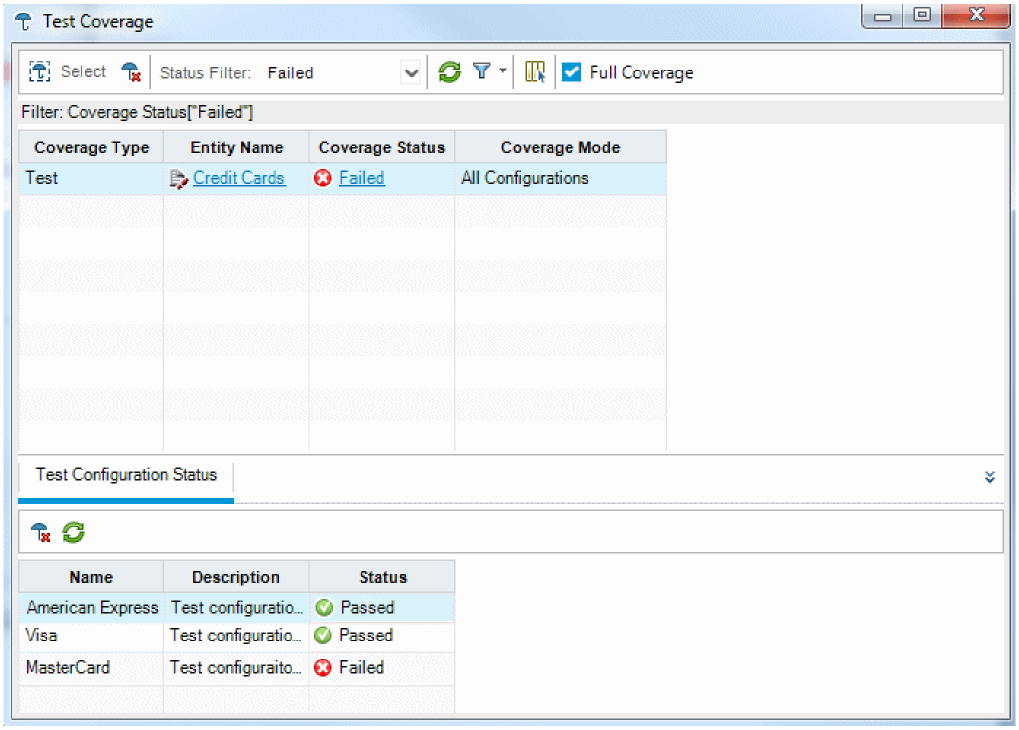


This graph displays the coverage status for requirement
372 - Credit Cards
and its children according to the current filter.
Click a group in the graph to display its list of requirements.

This graph displays the test status for requirement
372 - Credit Cards
Click a group in the graph to display its list of tests.

   c. This pie chart graphically displays the full test coverage for the requirement, grouped according to test status.

d. Click the pie chart to open the Test Coverage dialog box.



Status Filter: Failed

☑ Full Coverage

Filter: Coverage Status["Failed"]

Coverage Type	Entity Name	Coverage Status	Coverage Mode
Test	Credit Cards	Failed	All Configurations

Name	Description	Status
American Express	Test configuratio...	Passed
Visa	Test configuratio...	Passed
MasterCard	Test configuraito...	Failed

The dialog box shows the failed test together with the test run status of each test configuration.

e. Close the Test Coverage dialog box.

4. Close the Coverage Analysis dialog box.

   Click the **Close** button.

Viewing Coverage Progress

You were previously introduced to the Progress tab in the Releases module (see "View releases and cycles").

In this exercise, you will view the graphs and statistics that now reflect the results of tests you ran in previous exercises.

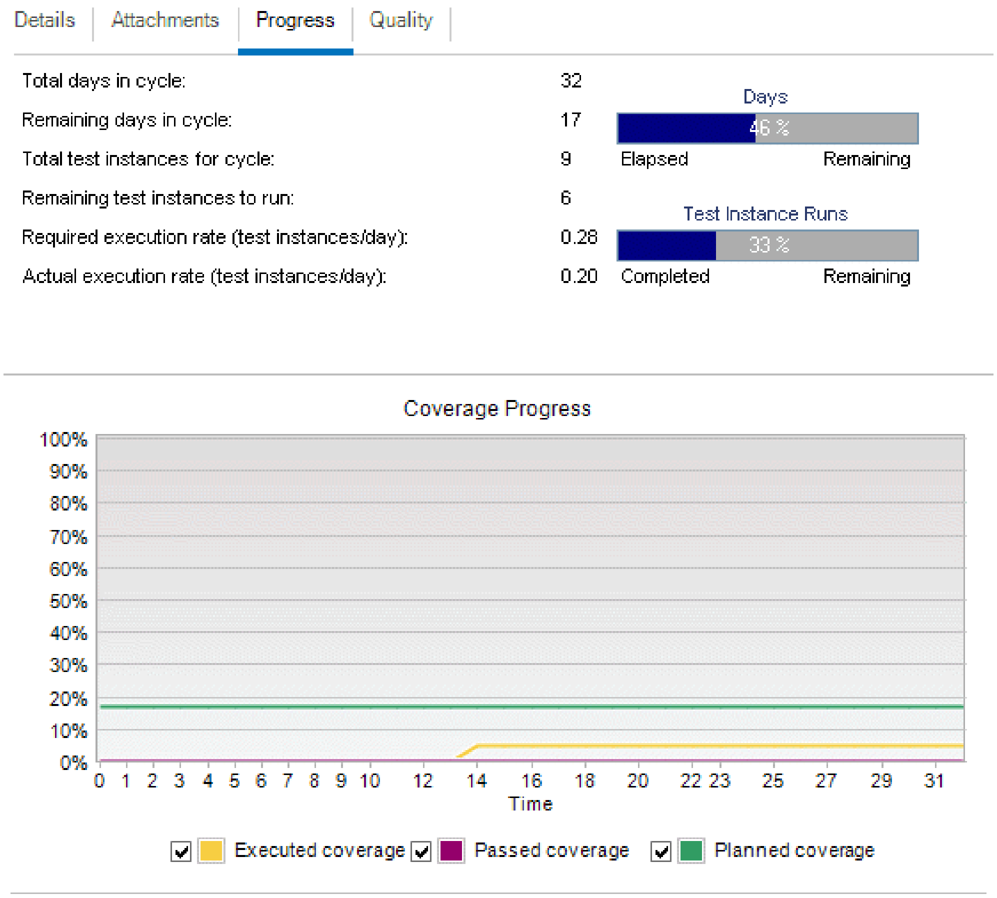
To view coverage progress:

1. Display the Releases module.

   On the ALM sidebar, under **Management**, select **Releases**.

2. View the effect of your test runs on the Progress tab.

   In the releases tree, expand the release folder **Service Packs** and the release **Service Pack 1**. Select **Cycle 1 - New Features** and click the **Progress** tab.



Total days in cycle:	32
Remaining days in cycle:	17
Total test instances for cycle:	9
Remaining test instances to run:	6
Required execution rate (test instances/day):	0.28
Actual execution rate (test instances/day):	0.20

The top pane shows information such as the total and remaining days in the cycle, total test instances for the cycle, and actual and remaining test instances to run.

The bottom pane displays the Coverage Progress graph. **Planned coverage** indicates the percentage of tests planned to be run each day within a cycle. **Executed coverage** indicates the percentage of tests that ran each day within a cycle. **Passed coverage** indicates the percentage of tests that ran successfully each day within a cycle.

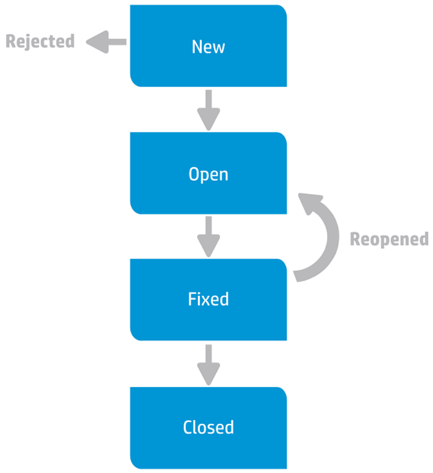
Adding and Tracking Defects

Locating and repairing defects is an essential phase in application development. Defects can be detected and submitted by users in all stages of the application lifecycle management process. Using ALM, you can submit defects detected in the application and track them until they have been repaired and retested.

In this lesson, you will learn about:

How to Track Defects

When you submit a defect to an ALM project, it is tracked through these stages: New, Open, Fixed, and Closed. A defect may also be Rejected or it may be Reopened after it is fixed.



When you initially report the defect to an ALM project, it is assigned the status **New**, by default. A quality assurance or project manager reviews the defect and determines whether or not to consider the defect for repair. If the defect is denied, it is assigned the status **Rejected**. If the defect is accepted, the quality assurance or project manager determines a repair priority, changes its status to **Open**, and assigns it to a member of the development team. A developer repairs the defect and assigns it the status **Fixed**. You then retest the application, making sure that the defect does not recur.

If the defect recurs, the quality assurance or project manager assigns it the status **Reopened**. If the defect is repaired, the quality assurance or project manager assigns it the status **Closed**.

Adding New Defects

You can add a new defect to an ALM project at any stage of the application lifecycle management process. In this exercise, you will submit a defect connected to the **Flight Confirmation** test.

To add a defect:

1. Open the ALM_Demo project.

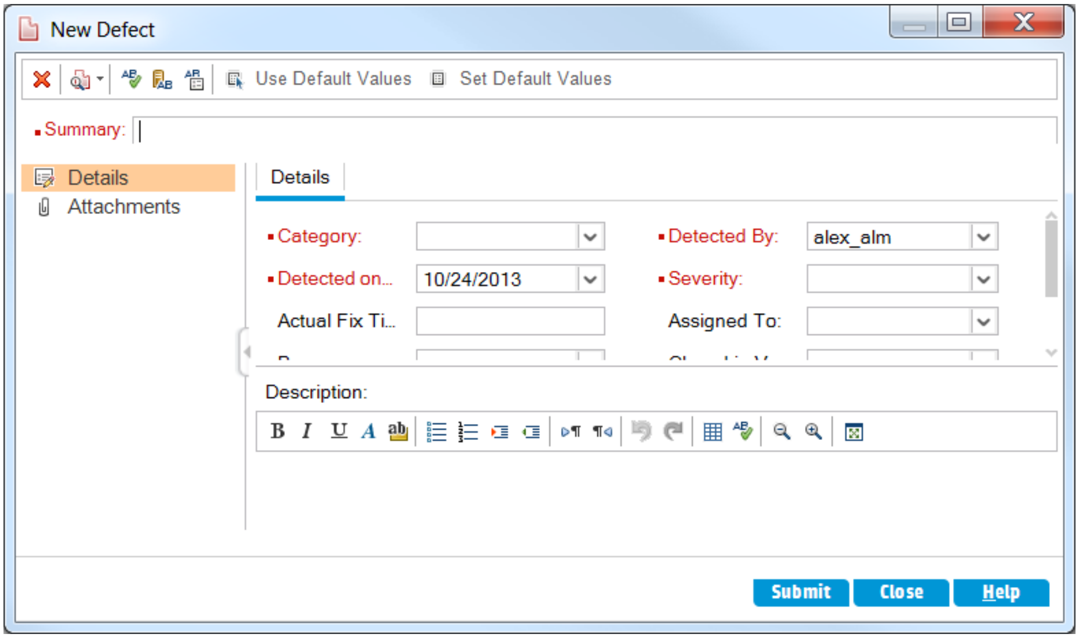
   If the **ALM_Demo** project is not already open, log in to the project. For details, see "Start" on page 10.

2. Display the Defects module.

   On the ALM sidebar, select **Defects**. The Defects Grid displays defect data in a grid. Each row in the grid displays a separate defect record.

3. Open the New Defect dialog box.

   Click the **New Defect** button. The New Defect dialog box opens.

   New Defect

   Summary:

   Category:

   Detected By: alex_alm

   Detected on... 10/24/2013

   Severity:

   Actual Fix Ti...

   Assigned To:

   Description:

4. Describe the defect.

Enter the following data in the specified fields. Scroll down the dialog box as necessary.

Summary: Missing information on Flight Confirmation page.

Category: Defect

Severity: 2-Medium

Subject: Flight Reservation > Flight Confirmation

Description: The defect was detected on the Flight Confirmation page. Passenger details and meal preferences are not displayed

5. Determine the cycle in which the defect was detected.

   a. In the **Detected in Cycle** box, click the arrow. The releases tree opens. Expand the tree. In the **Mercury Tours Application** releases folder, in the **Release 10.5** release, select the **Cycle 1 - New Features** cycle.

   b. Click **OK** to close the releases tree.

      You can see that ALM automatically assigns the value **Release 10.5** to the **Detected in Release** field. This is because **Cycle 1 - New Features** is part of the **Release 10.5** release.

6. Attach the URL address for the Mercury Tours page where the defect was detected.

   a. On the New Defect dialog box sidebar, click **Attachments**. The Attachment page opens.

   b. Click the **URL** button. The Attach Uniform Resource Locator (URL) dialog box opens.

   c. Type the URL address of the Mercury Tours page.

   d. Click **OK**. A link to the Mercury Tours page is displayed above the **Description** box.

7. Add the defect to the ALM project.

   a. Click the **Submit** button. The defect is added to the Defects Grid.

   b. Click **Close** to close the New Defect dialog box.

Matching Defects

Identifying matching defects enables you to eliminate duplicate or similar defects in your project. Each time you add a new defect, ALM stores lists of keywords from the **Summary** and **Description** fields. When you search for similar defects, keywords in these fields are matched against other defects. Note that keywords must be more than two characters long, and letter case does not affect your results.

In this exercise, you will match defects by comparing a selected defect with all other existing defects in the **ALM_Demo** project.

To match defects:

1. Make sure that the Defects module is displayed.

   If the Defects module is not displayed, on the ALM sidebar, select Defects.

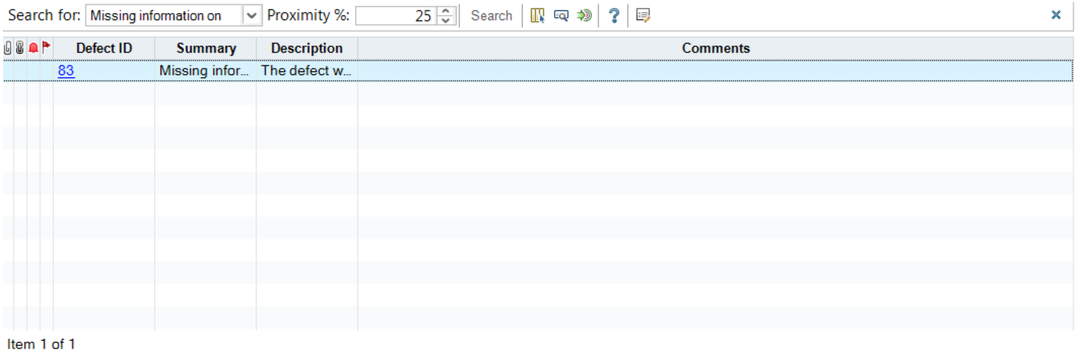
2. Select the Defect.

   a. If a filter is applied to the grid, click the **Set Filter/Sort** arrow and choose **Clear Filter/Sort** to clear the filter.

   b. In the Defects Grid, select the defect you added in "Adding New Defects" on page 99.

3. Find similar defects.

   Click the **Find Similar Defects** button. The results are displayed in the Similar Defects pane at the bottom of the window, sorted by the percentage of detected similarity. Note that there are no duplicate defects in the project for the selected defect.



Search for: Missing information on | Proximity %: 25

Defect ID	Summary	Description	Comments
83	Missing infor...	The defect w...	

Item 1 of 1

Close the Similar Defects pane.

Updating Defects

Tracking the repair of defects in a project requires that you periodically update defects. You can do so directly in the Defects Grid or in the Defect Details dialog box. The ability to update some defect fields depends on your permission settings. After you have updated defects, you can view the quality status of your release in the Releases module.

In this exercise, you will update your defect information by changing the severity of a defect, assigning the defect to a cycle, and adding a comment. You will then see how updating defects is reflected in the Quality tab of the Releases module.

In addition, you will practice updating multiple records simultaneously.

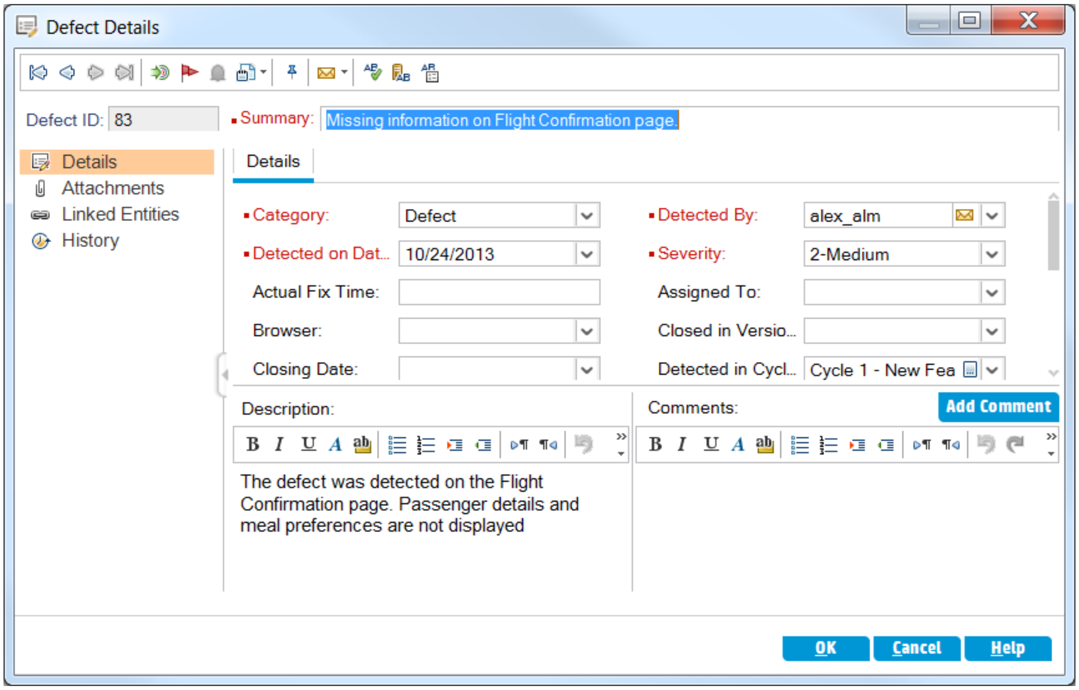
To update defects:

1. Make sure that the Defects module is displayed.

   If the Defects module is not displayed, on the ALM sidebar, select Defects.

2. Open the Defect Details dialog box.

   In the Defects Grid, select the defect you added in "Adding New Defects" on page 99. Click the **Defect Details** button. The Defect Details dialog box opens.



Defect Details

Defect ID: 83

Summary: Missing information on Flight Confirmation page.

Details

Category:	Defect	Detected By:	alex_alm
Detected on Dat...	10/24/2013	Severity:	2-Medium
Actual Fix Time:		Assigned To:	
Browser:		Closed in Versio...	
Closing Date:		Detected in Cycl...	Cycle 1 - New Fea

Description:

The defect was detected on the Flight Confirmation page. Passenger details and meal preferences are not displayed

Comments:

3. Change the severity level of the defect.

   In the Severity box, select 5-Urgent.

4. Determine the cycle in which the defect will be fixed.

   a. In the **Target Cycle** box, click the arrow. The releases tree opens. Expand the tree. In the **Mercury Tours Application** releases folder, in the **Release 10.5** release, select the **Cycle 4 a - Full cycle**.

   b. Click **OK** to close the releases tree.

      You can see that ALM automatically assigns the value **Release 10.5** to the **Target Release** field. This is because the **Cycle 4 a - Full cycle** is part of the Release 10.5 release.

5. Add a new comment to explain the change in the severity level.

   a. Click the **Add Comment** button. A new section is added to the **Comments** box, displaying your user name and the current date.

   b. Type: This defect should be fixed in the next service pack.

6. View the Attachments.

   Click **Attachments** on the sidebar. Note that the URL attachment is listed.

7. View Linked Entities.

   Click **Linked Entities** on the sidebar to view entities that have been linked to the defect. Linked entities can include requirements, tests, test sets, test instances, runs, run steps, and other defects. There are currently no linked entities. You will learn how to link a defect to a test in "Linking Defects to Tests" on page 106.

8. View the History.

   Click **History** on the sidebar to view the history of changes made to the defect. For each change, the grid displays the date and time of the change and the name of the user who made the change. You can expand a change to view a list of fields modified during the change. For each field, the grid displays the old value and the new value.

9. Close the Defect Details dialog box.

   Click **OK** to exit the dialog box.

10. Update defects directly in the Defects Grid.

    a. In the Defects Grid, select defect 35.

    b. Click the **Detected in Cycle** box and click the arrow. In the releases tree, expand the release folder **Mercury Tours Application** and the release **Release 10.5**. Select the **Cycle 1 - New Features** cycle and click **OK**.

    c. Drag the horizontal scroll bar to the right until the **Target Cycle** column comes into view. Click the **Target Cycle** box, and click the arrow. In the releases tree, expand the release folder **Mercury Tours Application** and the release **Release 10.5**. Select the **Cycle 1 - New Features** cycle and click **OK**.

11. Update multiple records simultaneously.

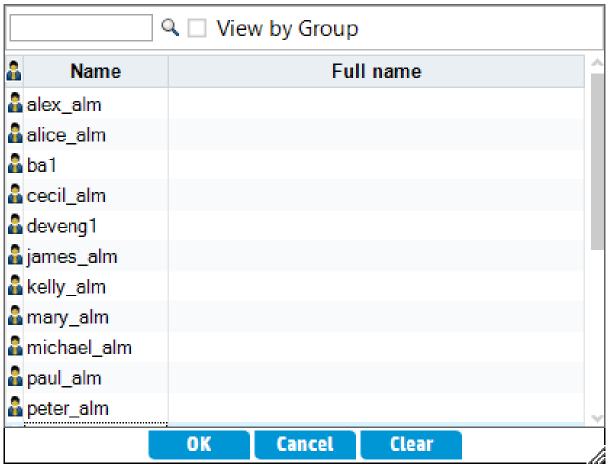
    a. In the Defects Grid, select defect 35.

    b. Hold down the CTRL key, and select defect 36. Now both defects 35 and 36 are selected.

c. Select **Edit > Update Selected**. The Update Selected dialog box opens.

d. Click the arrow adjacent to the **Update Field** box, and select **Assigned To**. Notice that the **Value** box displays the **Assigned To** value of the defect you selected last.

e. Click the arrow adjacent to the **Value** box. The users list opens.



Name	Full name
alex_alm	
alice_alm	
ba1	
cecil_alm	
deveng1	
james_alm	
kelly_alm	
mary_alm	
michael_alm	
paul_alm	
peter_alm	

f. In the users list, select **michael_alm**, and click **OK**.

g. In the Update Selected dialog box, click **Update**.

h. After the update completes, an Information dialog box displays a summary of the update. Click **OK**, then close the Update Selected dialog box.

12. View the number of defects opened in **Cycle 1 - New Features** on the Quality tab.

    a. On the ALM sidebar, under **Management**, select **Releases**. On the Releases tree, expand the release folder **Mercury Tours Application** and the release **Release 10.5**, and select the cycle **Cycle 1 - New Features**. Click the **Quality** tab.

    b. In the **Defect Opening Rate** graph, you can see the defects detected in **Cycle 1 - New Features** according to defect severity.

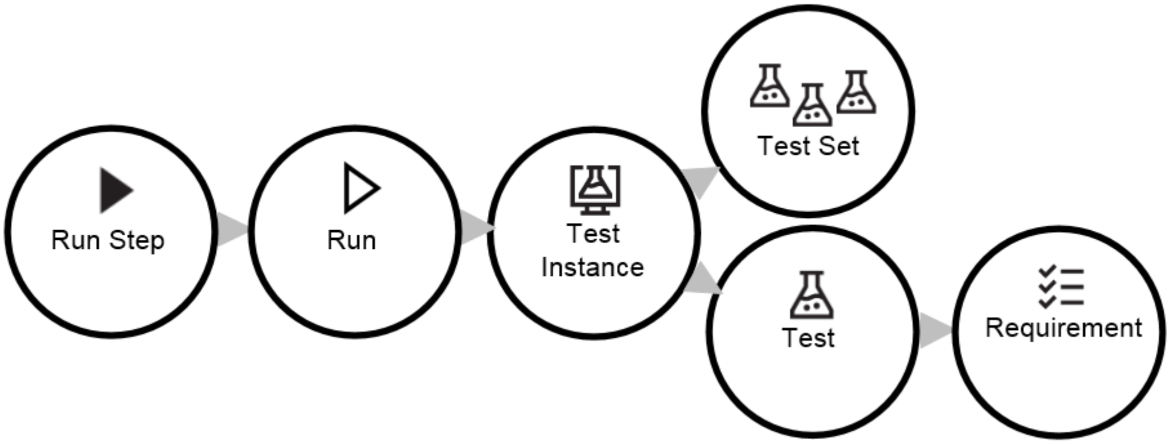
    c. In the **Outstanding Defects** graph, you can see the outstanding defects in **Cycle 1 - New Features** according to defect status.

Linking Defects to Tests

You can link a test in your test plan to a specific defect in the Defects Grid. This is useful, for example, when a new test is created specifically for a known defect. By creating linkage, you can determine if the test should be run based on the status of the defect. Note that you can also link the defect to other entities, such as requirements.

A defect can be linked directly or indirectly to an entity. When you add a defect link to an entity, ALM adds a direct link to this entity and indirect links to other related entities.

The following diagram illustrates the flow of indirect linkage:



For instance, when you link a defect to a run step, it adds an indirect link to its run, test instance, test set, and test. If the same test is covered by a requirement, an indirect link is also added to the requirement. Note that the indirect linkage is a one-directional flow. For example, if you link a defect to a run, it is not indirectly linked to its run steps.

In this exercise, you will link your defect to the **Flight Confirmation** test in the Test Plan module, and view the linked test in the Defects Grid.

To link a defect to a test:

1. Display the Test Plan module.

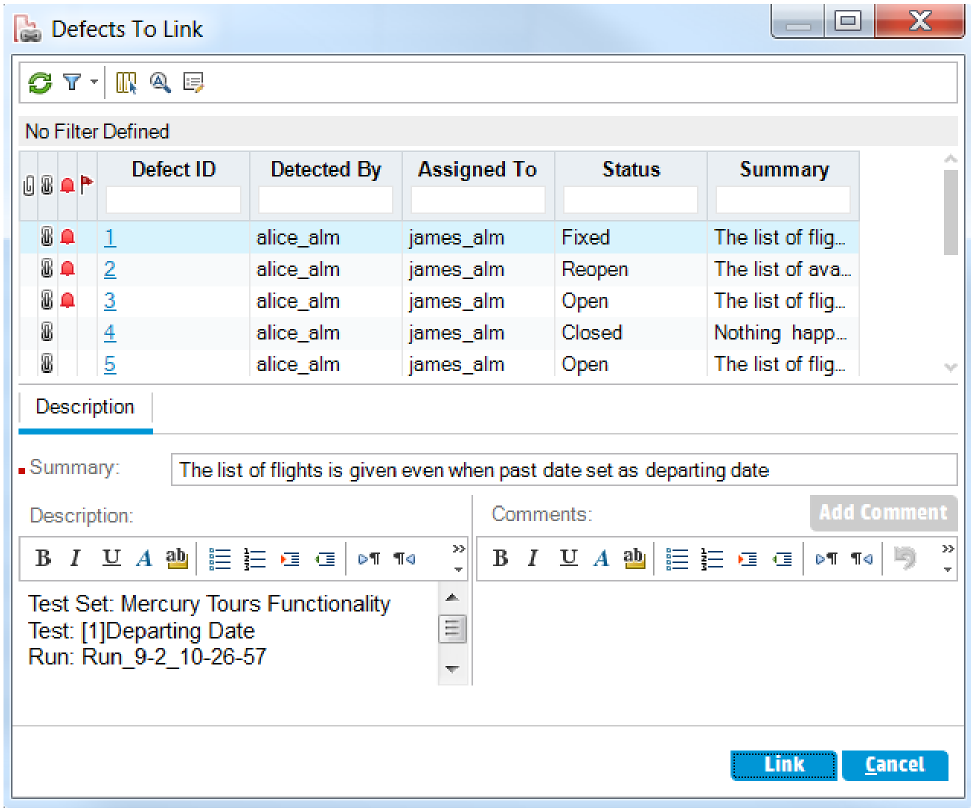
   On the ALM sidebar, under **Testing**, select **Test Plan**.

2. Select the Flight Confirmation test.

   In the test plan tree, under **Flight Reservation**, expand the **Flight Confirmation** test subject, and select the **Flight Confirmation** test. Click the **Linked Defects** tab.

3. Add a linked defect.

   a. In the Linked Defects tab, click the **Link Existing Defect** arrow and choose **Select**. The Defects to Link dialog box opens.



Defect ID	Detected By	Assigned To	Status	Summary
1	alice_alm	james_alm	Fixed	The list of flig...
2	alice_alm	james_alm	Reopen	The list of ava...
3	alice_alm	james_alm	Open	The list of flig...
4	alice_alm	james_alm	Closed	Nothing happ...
5	alice_alm	james_alm	Open	The list of flig...

Summary: The list of flights is given even when past date set as departing date

Description:
Test Set: Mercury Tours Functionality
Test: [1]Departing Date
Run: Run_9-2_10-26-57

   b. Select the defect you added in "Adding New Defects" on page 99 and click the **Link** button. Your defect is added to the Linked Defects grid.

> **Tip:** If you cannot find your defect in the Defects to Link dialog box, click the **Set Filter/Sort** arrow and choose **Clear Filter/Sort** to clear the filter that is applied to the grid.

4. View the linked test in the Defects Grid.

   a. On the ALM sidebar, select **Defects**.

   b. In the Defects Grid, click the defect ID of the defect you added in "Adding New Defects" on page 99. The Defect Details dialog box opens.

   c. Click **Linked Entities** on the sidebar, and select the **Others** tab. The **Flight Confirmation** test is linked to your defect.

   d. Click **OK** to close the Defect Details dialog box.

Creating Favorite Views

A favorite view is a view of an ALM window configured with the settings you applied to it. For example, in the Defects Grid, you may want to apply a filter to display only the defects that were detected by you, are assigned to you, or have the status “Not Closed”.

In this exercise, you will create a favorite view in the Defects Grid.

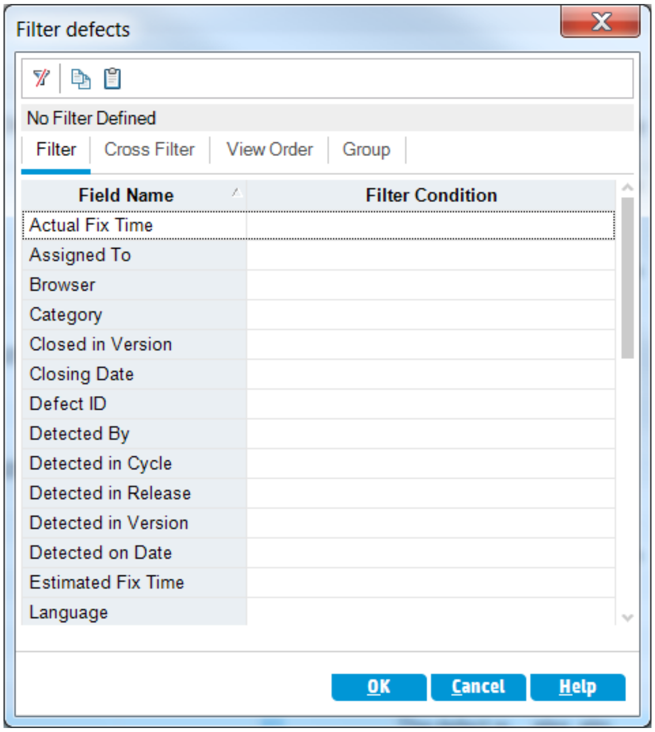
To create a favorite view:

1. Make sure that the Defects module is displayed.

   If the Defects module is not displayed, on the ALM sidebar, select **Defects**.

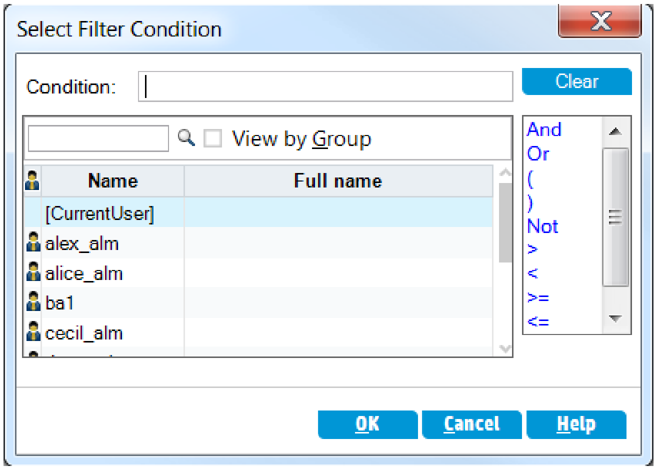
2. Define a filter to view defects you detected that are not closed.

a. Click the **Set Filter/Sort** button. The Filter defects dialog box opens.



Field Name	Filter Condition
Actual Fix Time	
Assigned To	
Browser	
Category	
Closed in Version	
Closing Date	
Defect ID	
Detected By	
Detected in Cycle	
Detected in Release	
Detected in Version	
Detected on Date	
Estimated Fix Time	
Language	

b. For the **Detected By** field, click the **Filter Condition** box. Click the arrow. The Select Filter Condition dialog box opens.



Name	Full name
[CurrentUser]	
alex_alm	
alice_alm	
ba1	
cecil_alm	

c. Under **Name**, select the **[CurrentUser]** variable, or select your ALM login name from the list. Click **OK** to close the Select Filter Condition dialog box.

   d. For the **Status** field, click the **Filter Condition** box. Click the arrow. The Select Filter Condition dialog box opens.

   e. In the right pane, click the logical expression **Not**.

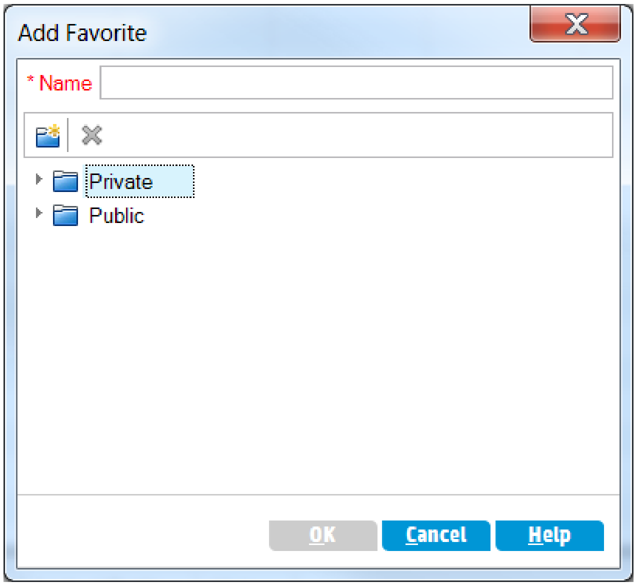
   f. In the left pane, select **Closed**.

   g. Click **OK** to close the Select Filter Condition dialog box.

   h. Click **OK** to apply your chosen filter. The Defects Grid displays the defects you detected that are not closed.

3. Add a favorite view.

   a. In the **Favorites** menu, select **Add to Favorites**. The Add Favorite dialog box opens.

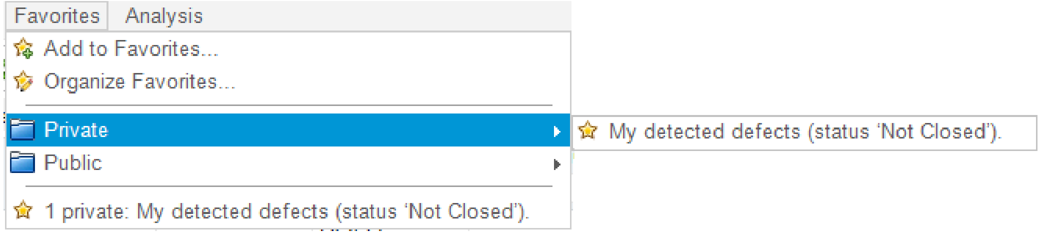


   b. In the **Name** box, type: My detected defects (status ‘Not Closed’).

   c. You can add a favorite view to either a **public** folder or a **private** folder. Views in the public folder are accessible to all users. Views in the private folder are accessible only to the person who created them. Select **Private**.

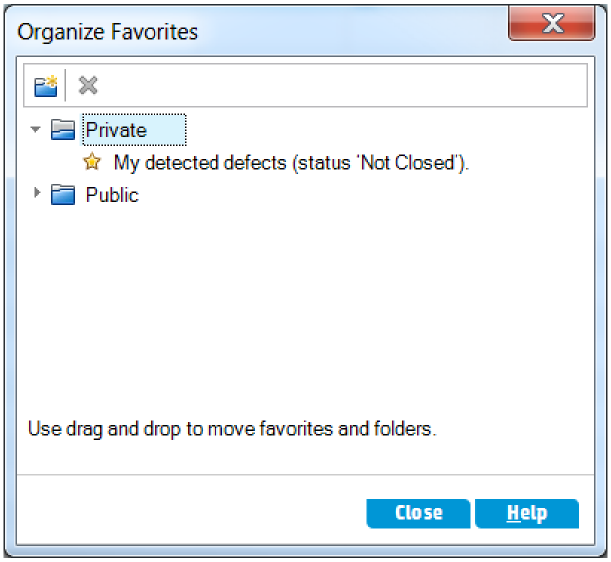
   d. Click **OK**. The new favorite view is added to your private folder, and is displayed in the list of recently used favorite views, located under the

Private and Public folders.


Favorites
Analysis
Add to Favorites...
Organize Favorites...
Private
Public
1 private: My detected defects (status 'Not Closed').
My detected defects (status 'Not Closed').

4. Organize favorites.

    a. In the **Favorites** menu, select **Organize Favorites**. The Organize Favorites dialog box opens.

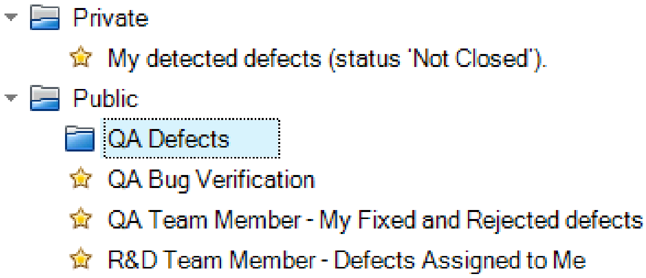
    

    Use drag and drop to move favorites and folders.

    b. Double-click the **Public** folder.

    c. Click the **New Folder** button. The New Favorite Folder dialog box opens.

    d. In the **Favorite Folder Name** box, type QA defects, and click **OK**. The **QA defects** folder is added as a sub-folder to the **Public** folder.

    
    Private
    My detected defects (status 'Not Closed').
    Public
    QA Defects
    QA Bug Verification
    QA Team Member - My Fixed and Rejected defects
    R&D Team Member - Defects Assigned to Me

e. Drag the following favorites into the **QA defects** folder:
   - **QA Bug Verification**
   - **QA Team Member - My Fixed and Rejected defects**

f. Click **Yes** to confirm, and close the Organize Favorites dialog box.

Alerting on Changes

You can instruct ALM to create alerts automatically and send emails to notify those responsible when certain changes occur in your project that may impact the application lifecycle management process. You can also add your own follow-up alerts.

To generate automatic notification alerts, your ALM project administrator must activate alert rules in Project Customization. Alert rules are based on associations you make in ALM between requirements, tests, and defects. When an entity in your project changes, ALM alerts any associated entities that may be impacted by the change. The alerts can be seen by all users. ALM also notifies the person responsible for the entity at the time of the change of any associated entities that may be impacted by the change.

ALM also enables you to add your own follow-up flag to a specific requirement, test, test instance, or defect to remind you to follow up on an issue. When the follow-up date arrives, ALM sends you an email reminder.

In this lesson, you will learn about:

Triggering an Alert

When a requirement, test, test instance, or defect in your project changes, ALM can notify those responsible for any associated entities. You can associate tests with requirements (see "Creating and Viewing Coverage" on page 42) and defects with other ALM entities (see "Linking Defects to Tests" on page 106). In addition, you can create traceability links between requirements. For details on creating traceability links between requirements, see "How to Use the Traceability Matrix" on page 1.

ALM can generate alerts for these changes:

What changed?	Which associated entities are flagged?	Who is notified?
Requirement (excluding change of **Direct Cover Status** and risk-based quality management fields)	Tests	Test designers
	Requirements	Requirement authors
Defect status changed to **Fixed**	Test instances	Responsible testers
Test ran successfully	Defects	Users assigned to defects

In this exercise, you will trigger alerts for tests by changing the associated requirement. You will modify the **View Reservations** requirement and then you will view the flagged tests.

To trigger an alert:

1. Display the requirements tree.

   a. On the ALM sidebar, under **Requirements**, select **Requirements**.

   b. In the Requirements module, select **View > Requirement Details**.

2. Select the requirement that you want to change.

   Under **Reservation Management**, select the **View Reservations** requirement.

3. View the associated tests.

   To view the tests that will be impacted by the change, click the **Test Coverage** tab. The tab displays the associated tests.

4. Change the priority of the View Reservations requirement.

   a. Click the **Details** tab.

   b. Click the down arrow adjacent to the **Priority** box and select **5-Urgent**.

      > **Note: Version Control**: If you are prompted to check out the requirement, click **OK**.

      This change causes ALM to generate alerts for the tests associated with the requirement. ALM also sends email notification to the designers of the associated tests.

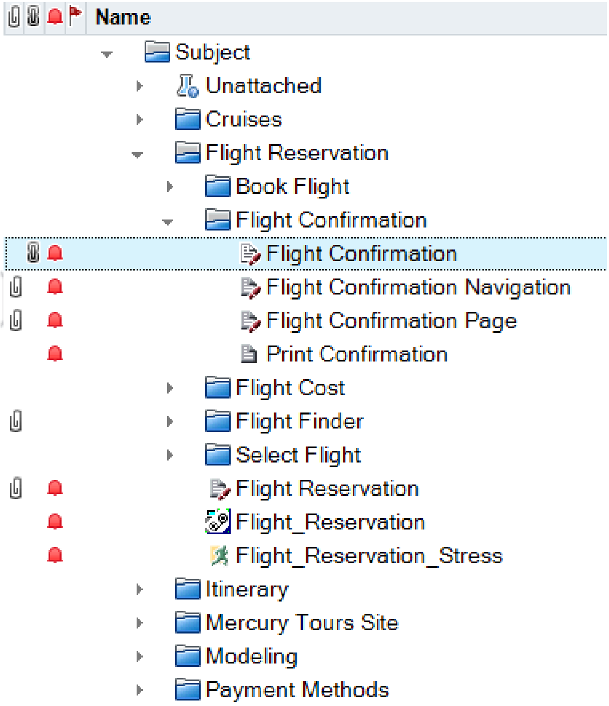
      > **Note: Version Control**: Check in the new version of the requirement. Right-click the requirement, and select **Versions > Check In**. Click **OK** to confirm.

5. View the alert for the Flight Confirmation test.

   a. In the **Test Coverage** tab, click the link to the **Flight Confirmation** test. The test is highlighted in the test plan tree.

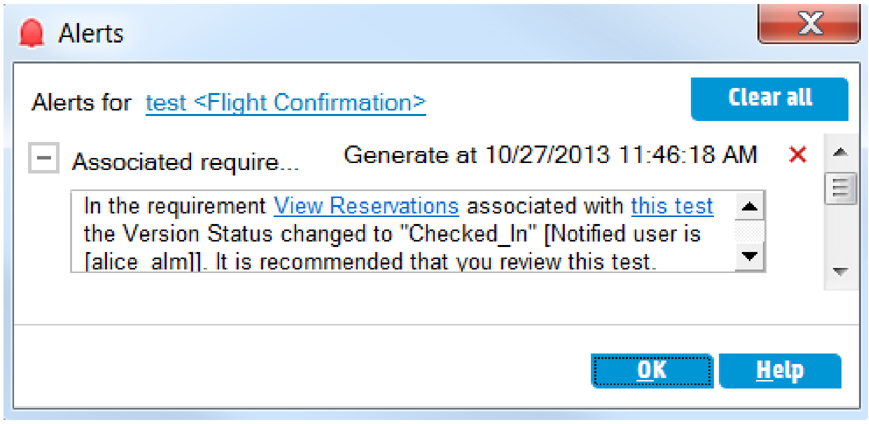
      > **Tip:** If you cannot find the test in the test coverage grid, click the **Set Filter/Sort** arrow and choose **Clear Filter/Sort** to clear the filter that is applied to the grid.

   b. Click the **Refresh All** button.



The **Flight Confirmation** test has an alert flag 🔔, indicating that a change was made to an associated requirement.

c. Click the **Alerts** flag for the **Flight Confirmation** test. The Alerts dialog box opens.



Alerts for test <Flight Confirmation>

Associated require... Generate at 10/27/2013 11:46:18 AM

In the requirement View Reservations associated with this test the Version Status changed to "Checked_In" [Notified user is [alice_alm]]. It is recommended that you review this test.

The alert indicates the requirement and the change that triggered the alert. It also indicates the name of the person to whom ALM sends email notification of this change.

> ! **Note: Version Control**: The alert states that the version status has changed to **Checked In**. It does not indicate which fields have been modified. You can then compare the new version with the previous version.

   d. Click **OK**.

6. View the alerts of the other associated tests.

   In the test plan tree, view the alerts of the following tests: **Itinerary > Itinerary Page** and **Itinerary > View Reservations > Review Reservations**.

Creating Follow Up Alerts

ALM enables you to add your own alerts to remind you to follow up on outstanding issues. In this exercise, you will add a follow-up flag to a defect whose status you want to check one week from today.

When you add a follow-up flag, ALM also adds an information bar that reminds you about the follow-up alert. When the follow-up date arrives, ALM sends you an email reminder, and changes the flag icon from gray to red.

Follow-up flags are specific to your user name, meaning that only you can see your follow-up alerts.

To create a follow up alert:

1. Display the Defects module.

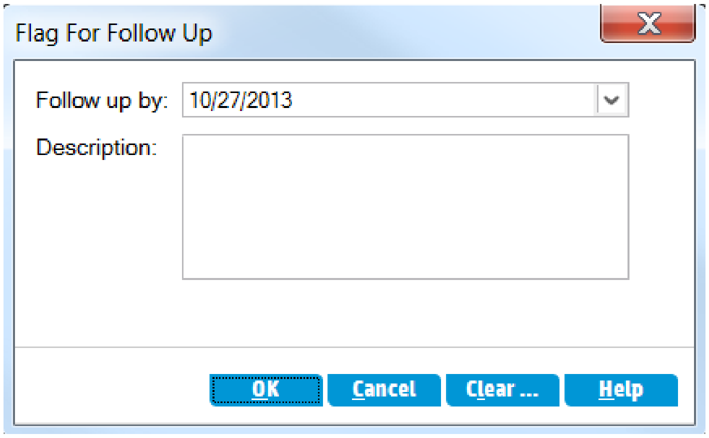
   On the ALM sidebar, select **Defects**.

2. Select the defect that you want to flag with a follow up reminder.

   In the Defects Grid, select a defect.

3. Create the follow up alert.

   a. Click the **Flag for Follow Up** button. The Flag For Follow Up dialog box opens.



Flag For Follow Up

Follow up by: 10/27/2013

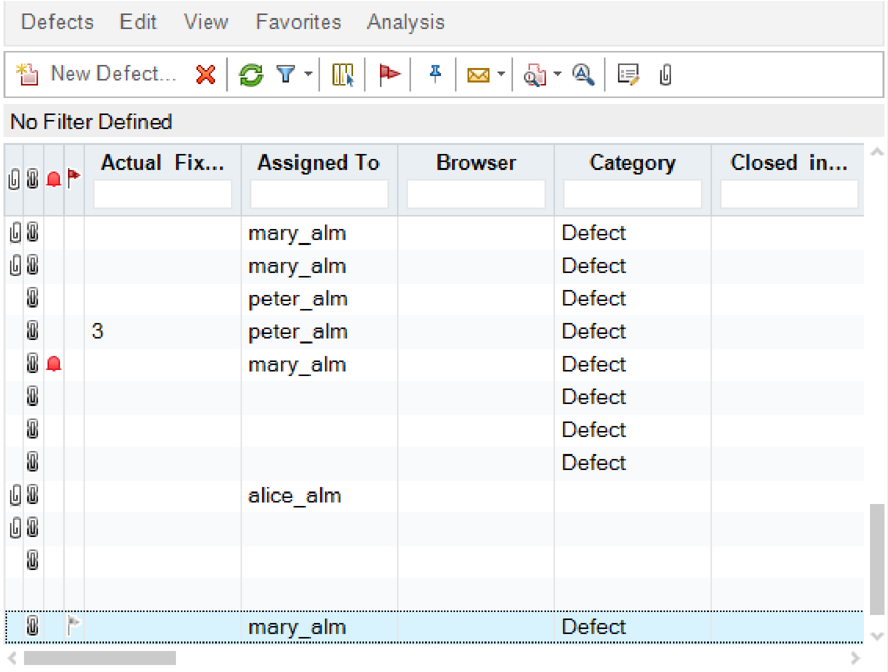
Description:

b. Perform the following:

Follow up by: Select the date one week from today.

Description: Type: Remind me about this defect on this date.

c. Click **OK**. The flag icon is added to the defect record.



No Filter Defined

Actual Fix...	Assigned To	Browser	Category	Closed in...
	mary_alm		Defect	
	mary_alm		Defect	
	peter_alm		Defect	
3	peter_alm		Defect	
	mary_alm		Defect	
			Defect	
			Defect	
			Defect	
	alice_alm			
	mary_alm		Defect	

Analyzing Data

Reports and graphs help you assess your application lifecycle management process. You can generate reports and graphs at any time during the process.

Project reports enable you to design and generate versatile reports containing information from the project.

You can create graphs or project reports either in the Analysis View module, or during your work in the Requirements, Test Plan, Test Lab, Business Components, or Defects modules. In either case, you can save the graphs and reports in the Analysis View module for future reference.

Using the Dashboard View module, you can also create dashboard pages that display multiple graphs side-by-side.

In this lesson, you will learn about:

Generating Project Reports

Project reports enable you to compile versatile reports of project data.

In a project report, you define sections and sub-sections, each listing records of a specified ALM entity.

To each report section, you assign a template that determines the fields and layout of the section. You also assign document and style templates that determine the overall report appearance.

In this exercise, you will create a report of tests with their linked defects.

To generate a project report:

1. Open the ALM_Demo project.

   If the **ALM_Demo** project is not already open, log in to the project. For details, see "Start" on page 10.

2. Display the Analysis View module.

   On the ALM sidebar, under **Dashboard**, select **Analysis View**.

3. Add a folder to the Private root folder.

   a. In the tree, select the **Private** folder.

   b. Click the **New Folder** button. The New Folder dialog box opens.

   c. Under **Folder Name**, type My_Analysis_Items.

   d. Click **OK**. The new folder is added as a sub-folder of the Private folder. The Details tab displays the folder name and the user who created the folder.

   e. In the **Description** tab, type This folder includes my demo analysis items.

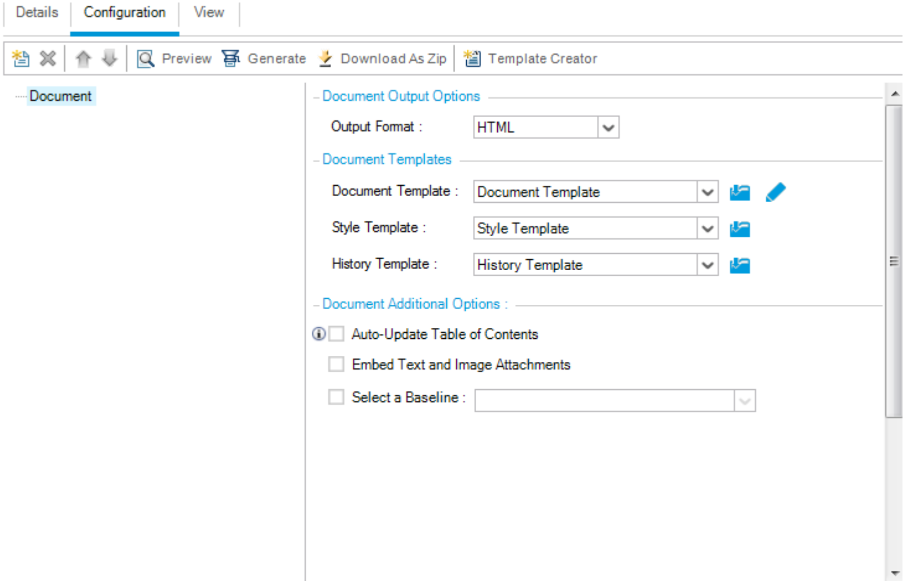
4. Create a project report.

   a. Click the **New Item** button and select **New Project Report**. The New Project Report dialog box opens.

b. In the **Project Report Name** box, type Reviewed tests and linked defects.

c. Click **OK**. A new project report is added to the folder you created. Notice that the 🗐 icon represents a project report.

The Configuration tab is selected.



Document Output Options

Output Format : HTML

Document Templates

Document Template : Document Template

Style Template : Style Template

History Template : History Template

Document Additional Options :

□ Auto-Update Table of Contents

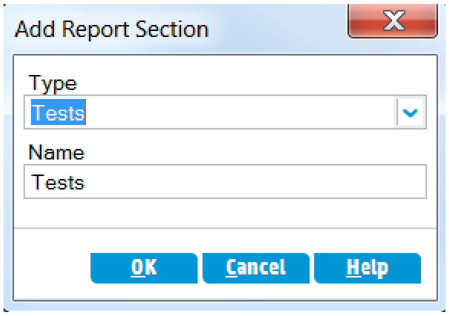
□ Embed Text and Image Attachments

□ Select a Baseline :

5. Add a main section to report.

a. Click the **Add Report Section** button. The Add Report Section dialog box opens.

b. Under **Type**, select **Tests**. The default section name Tests displays in the **Name** field.



Add Report Section

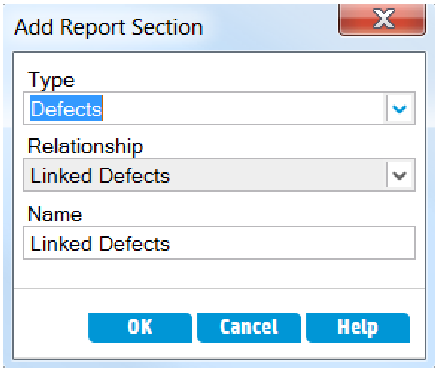
Type: Tests

Name: Tests

c. Click **OK**. The Tests section is added to the report tree, under the **Document** root folder.

6. Add a sub-section to the report.
   a. In the report tree, right-click the **Tests** section, and select **Add Report Section**.

      In the Add Report Section dialog box, you can select an entity related to the main section.
   b. Under **Type**, select **Defects**. The **Relationship** field describes the nature of the relationship between the tests and defects. The default section name Linked Defects displays in the **Name** field.

      Add Report Section

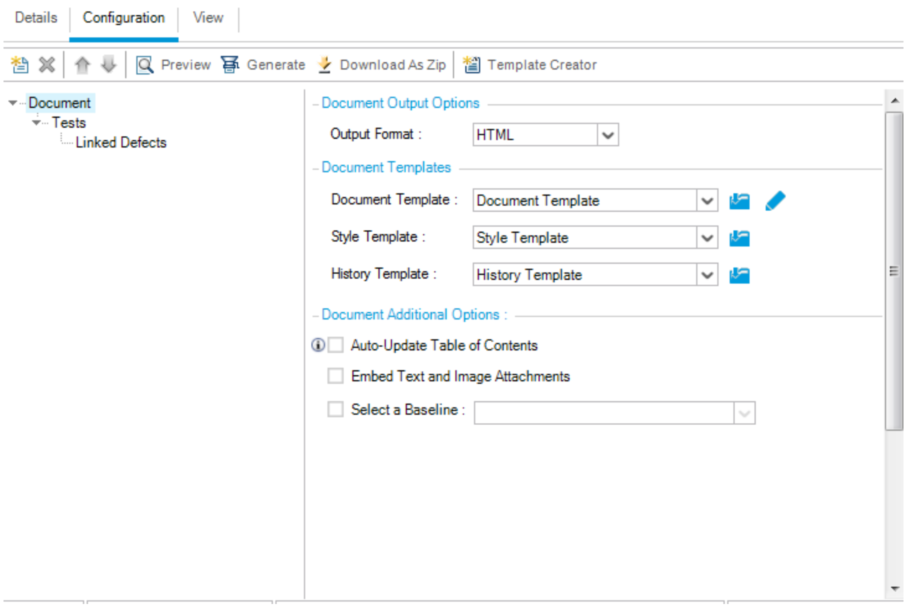
      Type: Defects

      Relationship: Linked Defects

      Name: Linked Defects

   c. Click **OK**. The **Linked Defects** section is added to the report tree, under the **Tests** section.
7. Configure document settings.

a. In the report tree, select the **Document** root node.



Document Output Options

Output Format : HTML

Document Templates

Document Template : Document Template

Style Template : Style Template

History Template : History Template

Document Additional Options :

□ Auto-Update Table of Contents

□ Embed Text and Image Attachments

□ Select a Baseline :

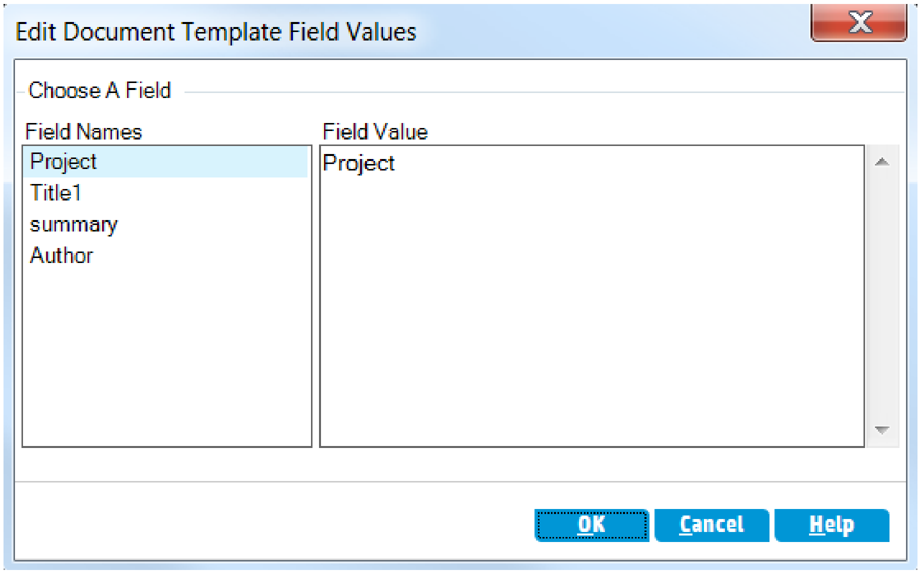
b. In the right pane, in the **Output Format** field, select **PDF**.

c. Under **Document Templates**, you assign templates that affect different areas of the report.

Template Type	Description
Document Template	Defines fields on the title page, headers and footers, page orientation, and other document layout settings.
Style Template	Defines formatting of report elements. For example, tables, headings, and paragraphs.
History Template	Defines the style in which history information is displayed in report sections.

Default templates are assigned to the different template types. Additional templates can be designed by the project administrator in Project Customization.

d. Click the **Edit Document Field Values** button alongside the Document Template field. The Edit Document Template Field Values dialog box opens.



The dialog box enables you to type values for fields in areas such as the title page, headers and footers, as defined in the document template.

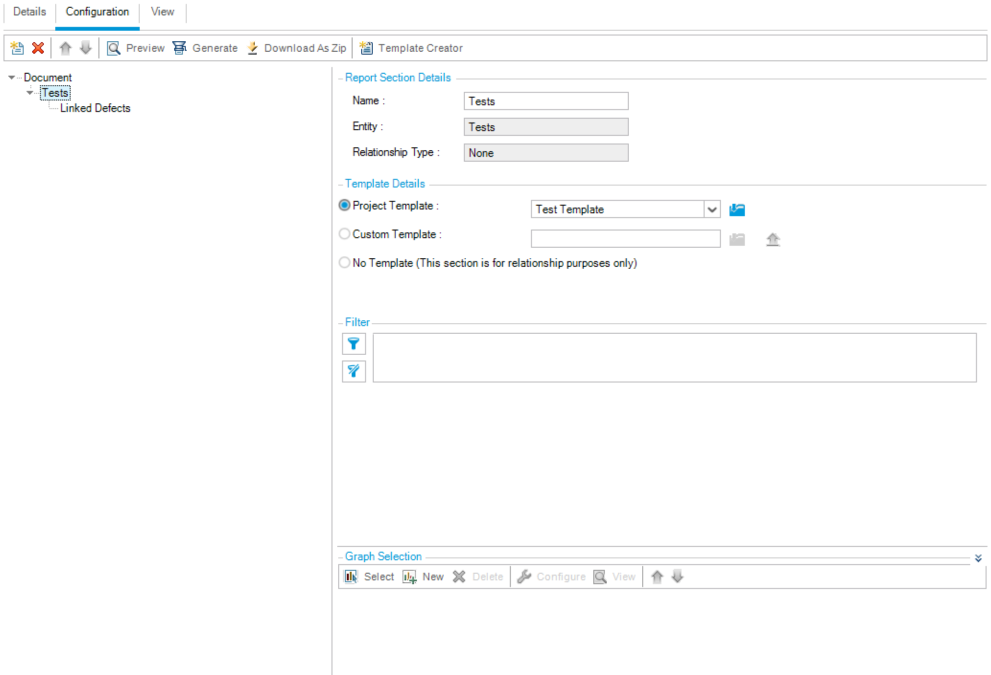
e. For each field name, type the following values:

Field Name	Field Value
Project	ALM Demo Project
Title1	Interim Report
summary	An interim report of reviewed tests and their linked defects.
Author	Alex ALM

f. Click **OK**.

8. Configure the Tests report section.

   a. In the report tree, select the **Tests** section.



Report Section Details

Name : Tests

Entity : Tests

Relationship Type : None

Template Details

● Project Template : Test Template

○ Custom Template :

○ No Template (This section is for relationship purposes only)

   b. Under **Report Section Details**, in the **Name** field, rename the section Reviewed Tests. The section name is used as the section title in the report.

   c. Under **Template Details**, make sure **Project Template** is selected.

   d. Click the arrow, and select **Test Template created by Alice**. The template you assign to the section determines the section format, and the entity fields displayed in the section.

   e. Under Filter, click the **Set Filter/Sort** button. The Filter Tests dialog box opens.

   f. Click the arrow adjacent to the **Reviewed** field.

   g. In the Select Filter Condition dialog box, select **Reviewed**, and click **OK**.

   h. Click **OK** to close the Filter Tests dialog box.

9. Configure the Linked Tests report section.

   a. Select the **Linked Defects** section.

   b. Click the arrow alongside **Project Template**, and select **Defect Tabular Template**. Tabular templates display records in rows of a table.

10. Generate the report.

    Click **Generate**. A dialog box opens, displaying the progress of the report generation.

    When the report is ready, a PDF reader opens, displaying the report results.

Generating Graphs

You can create graphs in the Analysis View module, that display data from the Requirements, Test Plan, Test Lab, Business Components, and Defects modules. You can also create predefined graphs directly from the Requirements, Test Plan, Test Lab, Business Components, and Defects modules. In either case, you can use a graph wizard to guide you through the stages of creating a graph.

You can create two types of graphs, entity graphs or business view graphs. Entity graphs display data directly from the various modules. Business view graphs are based on project entities, and ensure that only information that is relevant to a business consumer is contained in the graph.

After creating a graph, you can share the graph for viewing in a Web browser outside ALM.

This section includes:

Generating Entity Graphs in the Analysis View Module

In the Analysis View module, you can create graphs and configure them according to your specifications. You can also create a graph using the graph wizard. The graph wizard takes you through the steps involved in creating a graph and defining its settings.

You create graphs in either a public folder or a private folder. Graphs in a **public** folder are accessible to all users. Graphs in a **private** folder are accessible only to the user who created them.

In this exercise, you will use the graph wizard to generate an entity graph that summarizes the defects by status and priority level.

To generate an entity graph in the Analysis View module:

1. Make sure the Analysis View module is displayed.

   If the Analysis View module is not displayed, on the ALM sidebar, under **Dashboard**, select **Analysis View**.

2. Open the graph wizard.

   Click the **New Item** button and select **Graph Wizard**. The graph wizard opens to the Select Graph Type page.

3. Select the graph type.

   Select **Entity Graph**.

4. Select the entity type.

   a. Click **Next**. The Select Entity Type page opens.

   b. Under **Entity**, select **Defects**.

   c. Under **Graph Type**, make sure **Summary Graph** is selected.

5. Select the projects to include.

   a. Click **Next**. The Select Projects page opens.

   b. You can include data from multiple projects in the graph. For this exercise, we will use only the current project. Make sure **Use Current Project** is selected.

6. Define a filter to view defects with high to urgent priority.

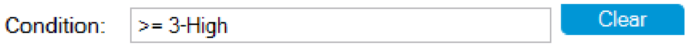
   a. Click **Next**. The Select Filter page opens.

   b. Under **Filter Selection**, select **Define a new filter**. Click the **Filter** button. The Filter defects dialog box opens.

   c. Click the down arrow adjacent to the **Priority** field. The Select Filter Condition dialog box opens.

   d. In the right pane, select the logical expression >=.

   e. In the left pane, select **3-High**. The condition appears as follows:



Condition: >= 3-High

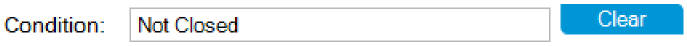
   f. Click **OK** to close the Select Filter Condition dialog box.

7. Define a filter to view defects that are not closed.

   a. Click the down arrow adjacent to the **Status** field. The Select Filter Condition dialog box opens.

   b. In the right pane, select the logical expression **Not**.

   c. In the left pane, select **Closed**. The condition appears as follows:



Condition: Not Closed

   d. Click **OK** to close the Select Filter Condition dialog box.

   e. Click **OK** to close the Filter defects dialog box.

8. Set the graph attributes.

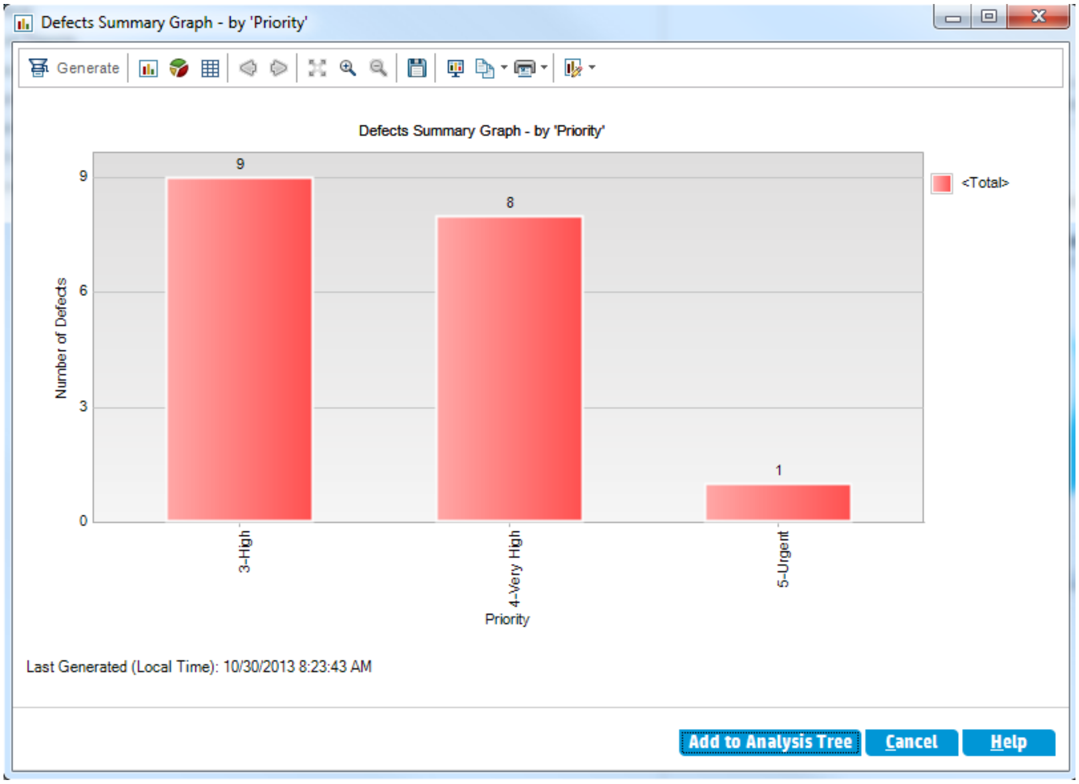
   a. Click **Next**. The Select Graph Attributes page opens.

   b. Under **Group By field**, make sure that it is set to **<None>**.

   c. Under **X-axis field**, select **Priority** to view the number of defects by priority.

9. Generate the Graph.

   Click **Finish**. The graph is displayed in the graph window.

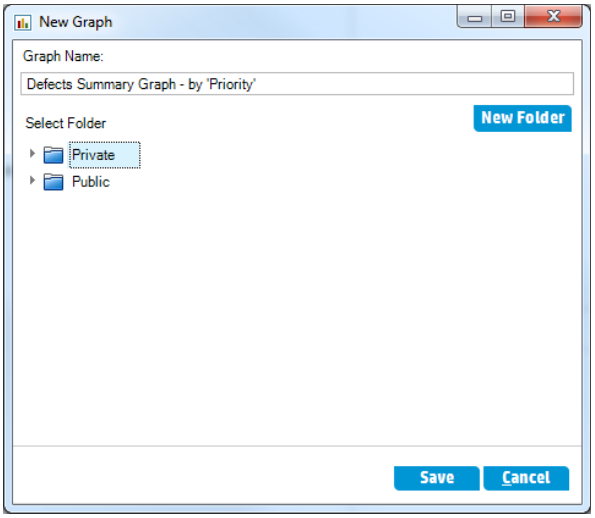


Last Generated (Local Time): 10/30/2013 8:23:43 AM

   The graph shows a summary of defects with **High** to **Urgent** priority, whose status is not **Closed**.

10. Save the graph in the Analysis View module.

a. Click **Add to Analysis Tree**. The New Graph dialog box opens.



Graph Name: Defects Summary Graph - by 'Priority'

Select Folder

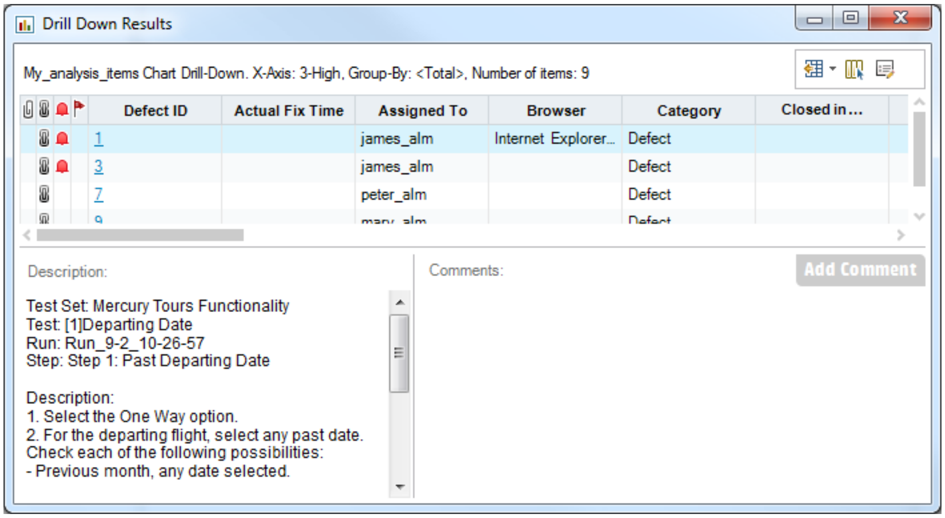
- Private
- Public

b. Expand the **Private** folder, and select **My_Analysis_Items**.

c. Click **Save**. The Defects Summary graph is saved in the analysis tree, and displayed in the View tab.

11. Display additional defect details.

a. Click a bar in the graph. The Drill Down Results dialog box opens and displays the defects that belong to the bar.



My_analysis_items Chart Drill-Down. X-Axis: 3-High, Group-By: <Total>, Number of items: 9

Defect ID	Actual Fix Time	Assigned To	Browser	Category	Closed in ...
1		james_alm	Internet Explorer...	Defect	
3		james_alm		Defect	
7		peter_alm		Defect	
9		mary_alm		Defect	

Description:

Test Set: Mercury Tours Functionality
Test: [1]Departing Date
Run: Run_9-2_10-26-57
Step: Step 1: Past Departing Date

Description:
1. Select the One Way option.
2. For the departing flight, select any past date.
Check each of the following possibilities:
- Previous month, any date selected.

Comments:

b. Close the Drill Down Results dialog box.

12. Display other graph views.

   a. Click the **Pie Chart** button to display the graph as a pie chart.
   b. Click the **Data Grid** button to display the data as a grid.

Generating Business View Graphs in the Analysis View Module

In this exercise, you will use the graph wizard to generate a business view graph that summarizes the requirements by status and priority level.

To generate a business view graph in the Analysis View module:

1. Make sure the Analysis View module is displayed.

   If the Analysis View module is not displayed, on the ALM sidebar, under **Dashboard**, select **Analysis View**.

2. Open the graph wizard.

   Click the **New Item** button and select **Graph Wizard**. The graph wizard opens to the Select Graph Type page.

3. Select the graph type.

   Select **Business View Graph**.

4. Select the business view.

   a. Click **Next**. The Select Business View page opens.
   b. Select **Requirements**.

5. Select the projects to include.

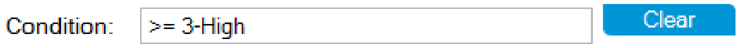
   a. Click **Next**. The Select Projects page opens.
   b. You can include data from multiple projects in the graph. For this exercise, we will use only the current project. Make sure **Use Current Project** is selected.

6. Define a filter to view requirements with high to urgent priority.

   a. Click **Next**. The Select Filter page opens.
   b. Click the down arrow in the Criteria column adjacent to the **Priority** field. The Select Filter Condition dialog box opens.

c. In the right pane, select the logical expression >=.

d. In the left pane, select **3-High**. The condition appears as follows:



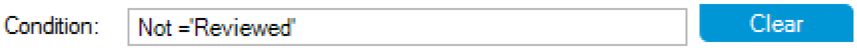
Condition: >= 3-High

e. Click **OK** to close the Select Filter Condition dialog box.

7. Define a filter to view defects that are not closed.

   a. Click the down arrow in the Criteria column adjacent to the **Reviewed** field. The Select Filter Condition dialog box opens.

   b. In the left pane, select **Not Reviewed**. The condition appears as follows:



Condition: Not ='Reviewed'

   c. Click **OK** to close the Select Filter Condition dialog box.

8. Set the graph attributes.

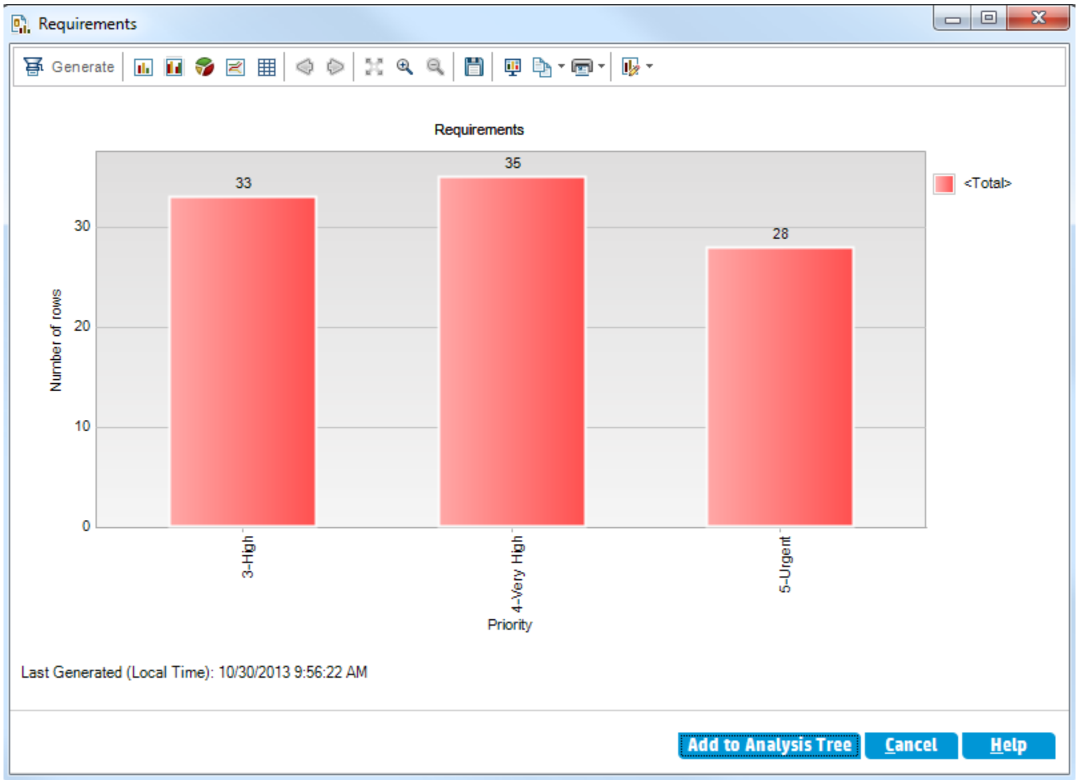
   a. Click **Next**. The Select Graph Attributes page opens.

   b. Under **X-axis**, select **Priority** and under Y-axis, select **Count** to view the number of requirements by priority.

   c. Under **Grouped By**, make sure that it is set to **<None>**.

9. Generate the Graph.

   Click **Finish**. The graph is displayed in the graph window.

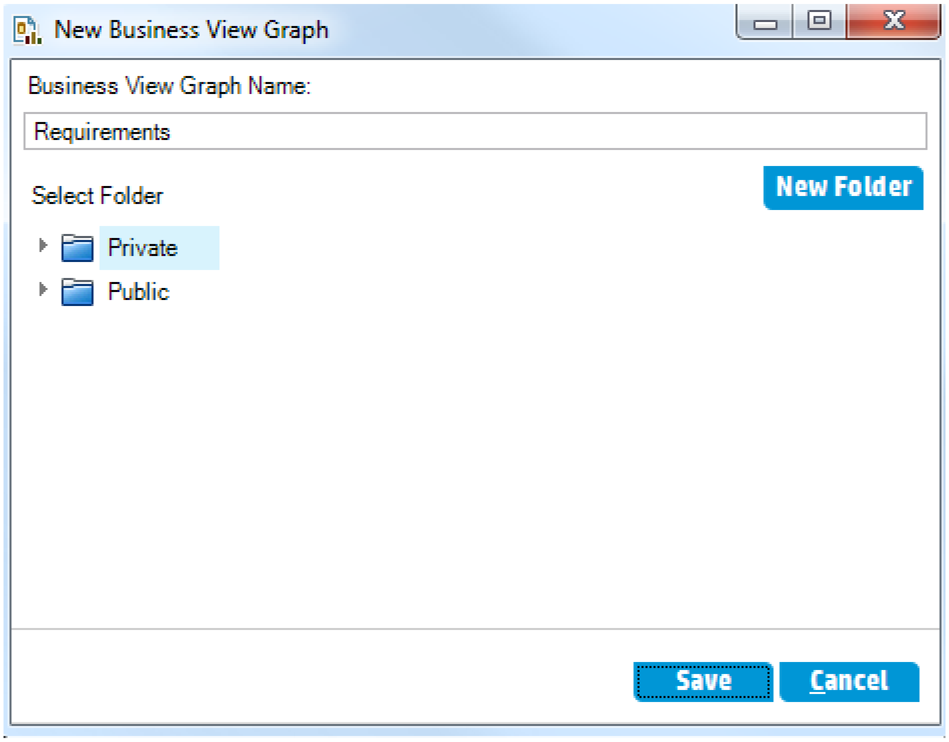


Last Generated (Local Time): 10/30/2013 9:56:22 AM

The graph shows a summary of requirements with **High** to **Urgent** priority that have not been reviewed.

10. Save the graph in the Analysis View module.

    a. Click **Add to Analysis Tree**. The New Business View Graph dialog box opens.



New Business View Graph

Business View Graph Name: Requirements

b. Expand the **Private** folder, and select **My_Analysis_Items**.

c. Click **Save**. The Requirements graph is saved in the analysis tree, and displayed in the View tab.

11. Display other graph views.

    a. Click the **Pie Chart** button to display the graph as a pie chart.

    b. Click the **Data Grid** button to display the data as a grid.

Generating Predefined Graphs

You can create predefined graphs during your work in the Requirements, Test Plan, Test Lab, Business Components, and Defects modules. Predefined graphs enable you to create several types of graphs in each module, using the existing module filter. You can use predefined graphs for one-time reference, or save them in the Analysis View module, where you can continue to configure their data and appearance.

In this exercise, you will generate a summary graph from the Defects module.

To generate a predefined graph:

1. Display the Defects module.

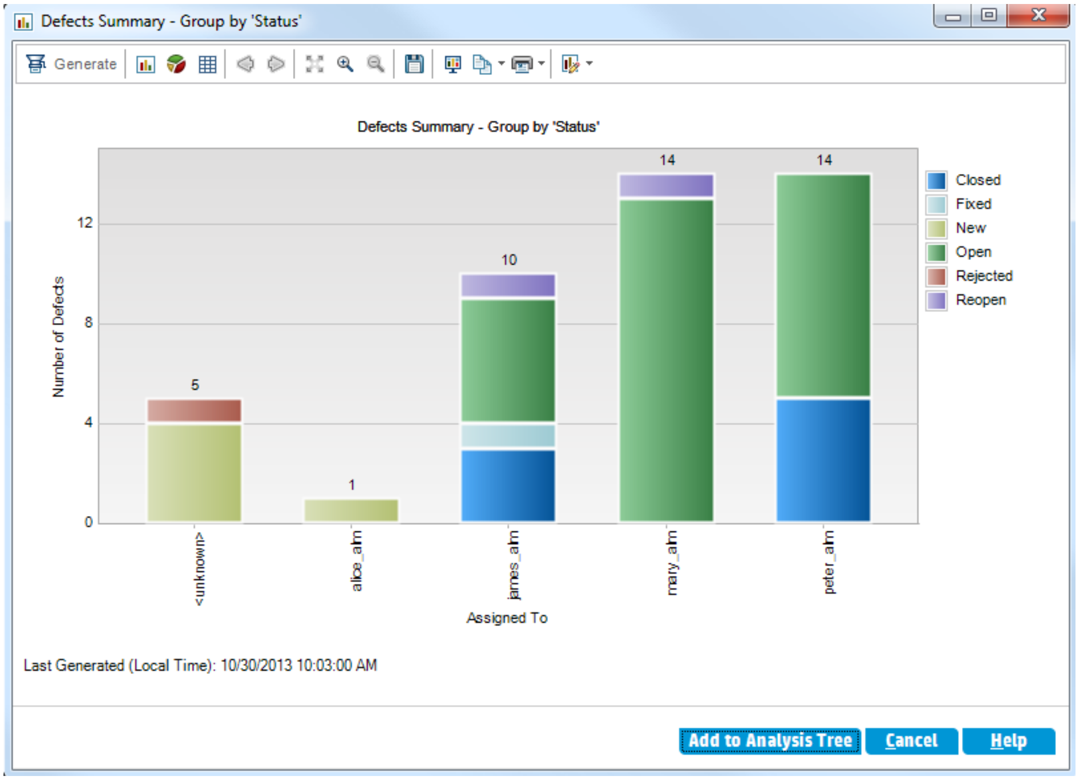
   On the ALM sidebar, select **Defects**.

2. Clear the Defects grid filter.

   Click the **Set Filter/Sort** arrow, and choose **Clear Filter/Sort** to clear the filter that is applied to the grid.

3. Generate a report.

   Choose **Analysis > Graphs > Defects Summary - Group by Status**. The graph window opens, displaying the selected graph.

   Last Generated (Local Time): 10/30/2013 10:03:00 AM

   The graph shows the number of defects that exist in your project, according to the people to whom they are assigned. Defects in the graph's columns are grouped according to their status.

4. Close the graph window.

   Click the **Cancel** button.

Sharing Graphs

You can allow ALM users or others to view read-only versions of graphs outside ALM. Each time you access the graph outside ALM, it displays the most up-to-date information.

In this exercise, you will share the Defects Summary graph you created in the Generating Entity Graphs in the Analysis View Module exercise.

To share a graph:

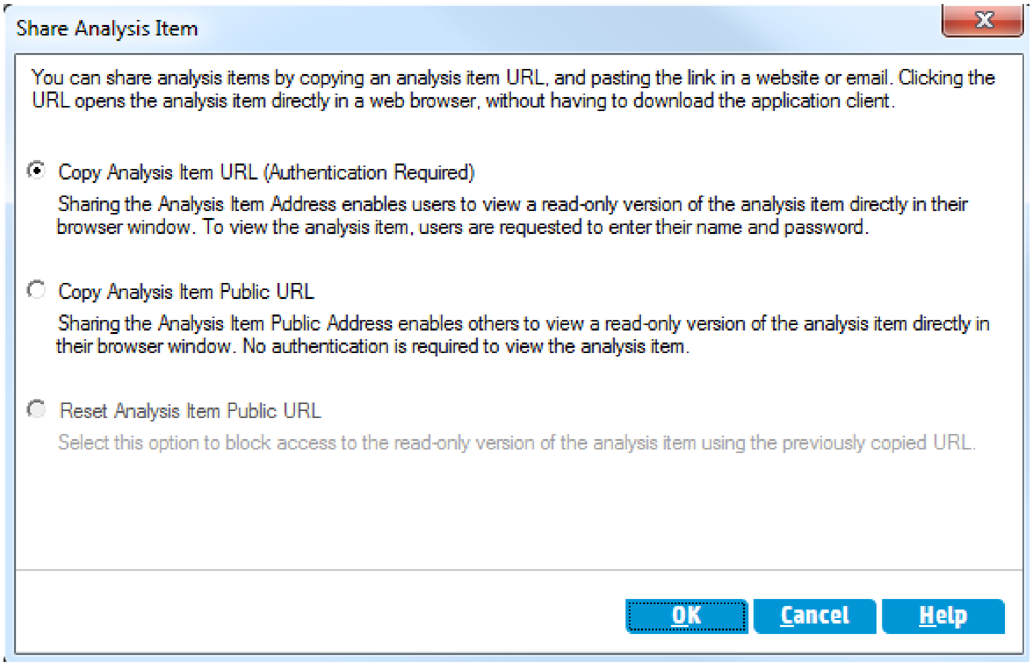
1. Display the Analysis View module.

   On the ALM sidebar, under **Dashboard**, select **Analysis View**.

2. Share the Defects Summary graph.

   a. In the **Private** folder, under **My_Analysis_Items**, select **Defects Summary Graph - Group by 'Priority'**.

   b. Right-click and select **Share Analysis Item**. The Share Analysis Item dialog box opens.



Share Analysis Item

You can share analysis items by copying an analysis item URL, and pasting the link in a website or email. Clicking the URL opens the analysis item directly in a web browser, without having to download the application client.

● Copy Analysis Item URL (Authentication Required)

Sharing the Analysis Item Address enables users to view a read-only version of the analysis item directly in their browser window. To view the analysis item, users are requested to enter their name and password.

○ Copy Analysis Item Public URL

Sharing the Analysis Item Public Address enables others to view a read-only version of the analysis item directly in their browser window. No authentication is required to view the analysis item.

○ Reset Analysis Item Public URL

Select this option to block access to the read-only version of the analysis item using the previously copied URL.

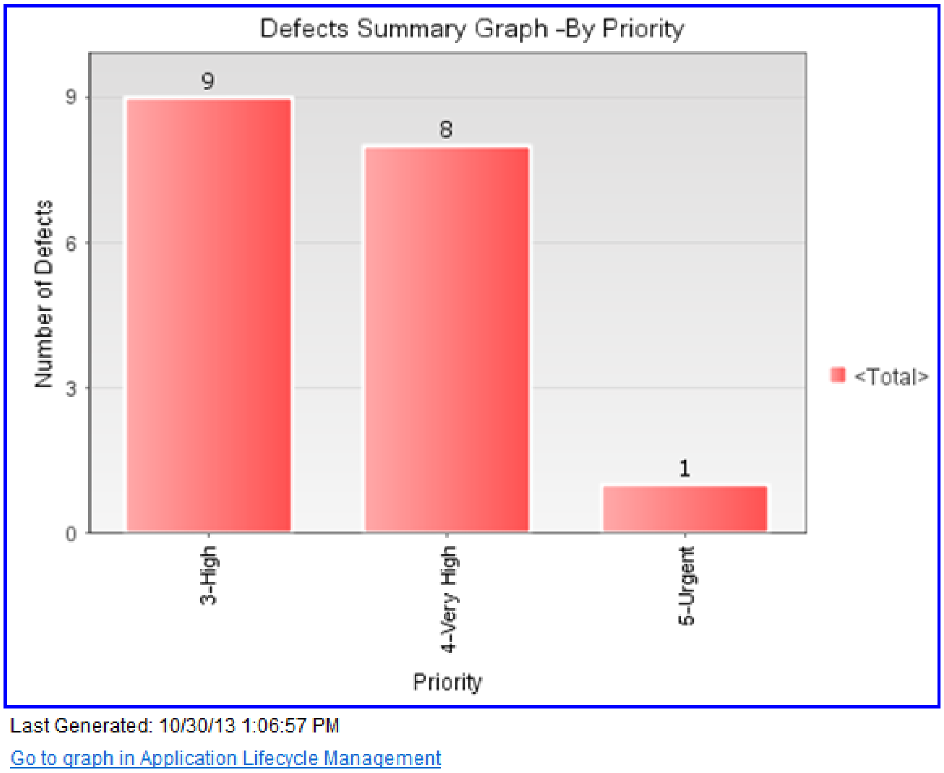
c. Select **Copy Analysis Item Public URL**. This option allows others to view the graph without entering ALM user credentials.

d. Click **OK**. A URL of the graph is saved to the clipboard.

3. View the graph outside ALM.

a. Open a Web browser window.

b. In the address bar, paste the clipboard content, and press ENTER. The graph is displayed in the Web browser.



Last Generated: 10/30/13 1:06:57 PM

Go to graph in Application Lifecycle Management

Below the graph, the **Last Generated** date and time is displayed, and a link to the item in ALM.

Generating Business View Excel Reports

You can create Excel reports based on business views. A business view is a data layer that exists on top of the database and which reflects only those project entity fields that represent information that is useful from a business perspective. Business views can be based on single entities, such an Baselines or Defects, while others can represent more complex relationships between entities, such as Defects With Linked Requirements.The reports are created and configured in Microsoft Excel, and uploaded to the Analysis View module.

In this exercise, you will use Microsoft Excel to generate a business view Excel report that summarizes the defects by status and priority level.

To generate a business view Excel report in the Analysis View module:

1. Make sure the Analysis View module is displayed.

   If the Analysis View module is not displayed, on the ALM sidebar, under **Dashboard**, select **Analysis View**.

2. Create a new business view Excel report.

   a. In the **Private** folder, select the **My_Analysis_Items** folder.

   b. Click the **New Item** button and select **New Business View Excel Report**. The New Business View Excel Report dialog box opens.

   c. In the **Business View Excel Report Name** box, type High Priority Open Defects.

   d. Click **OK**. A new business view Excel report is added to the folder you created. Notice that the icon represents a business view Excel report.

   The Configuration tab is selected.

3. Open Excel.

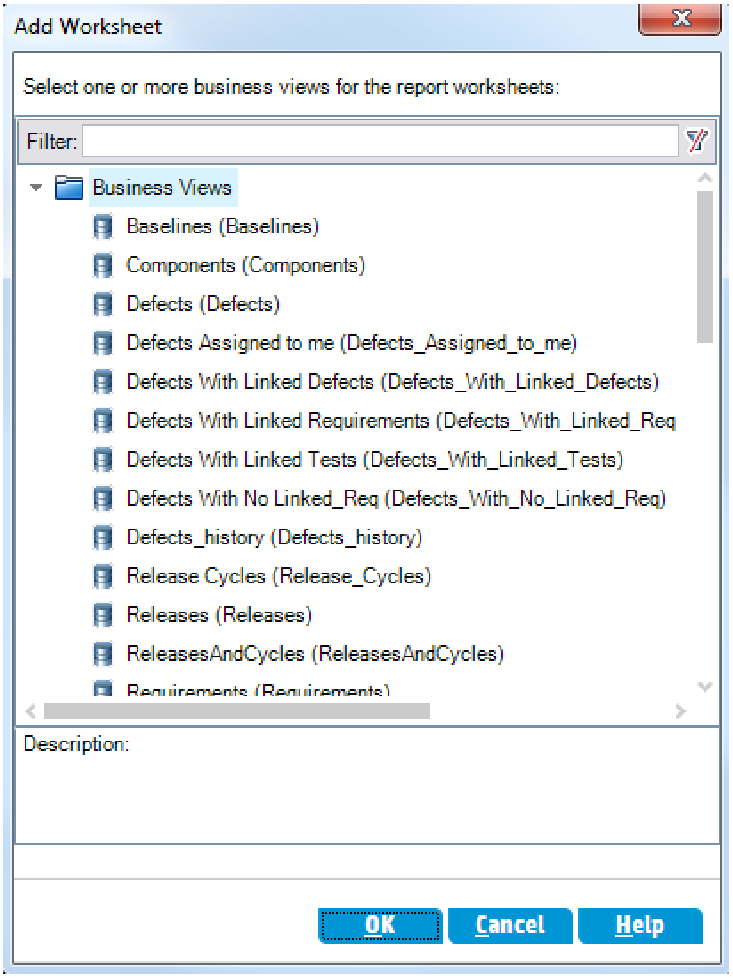
   a. Click **Install Addin**. The ALM Business Views Microsoft Excel Add-in is installed in Excel.

   b. Click **New Excel**. Microsoft Excel opens with the ALM tab in the ribbon.

   c. In the **ALM** tab, click **Login**. The ALM login window opens. Log in exactly as you logged in to ALM at the beginning of the tutorial.

4. Generate the business view report in Excel.

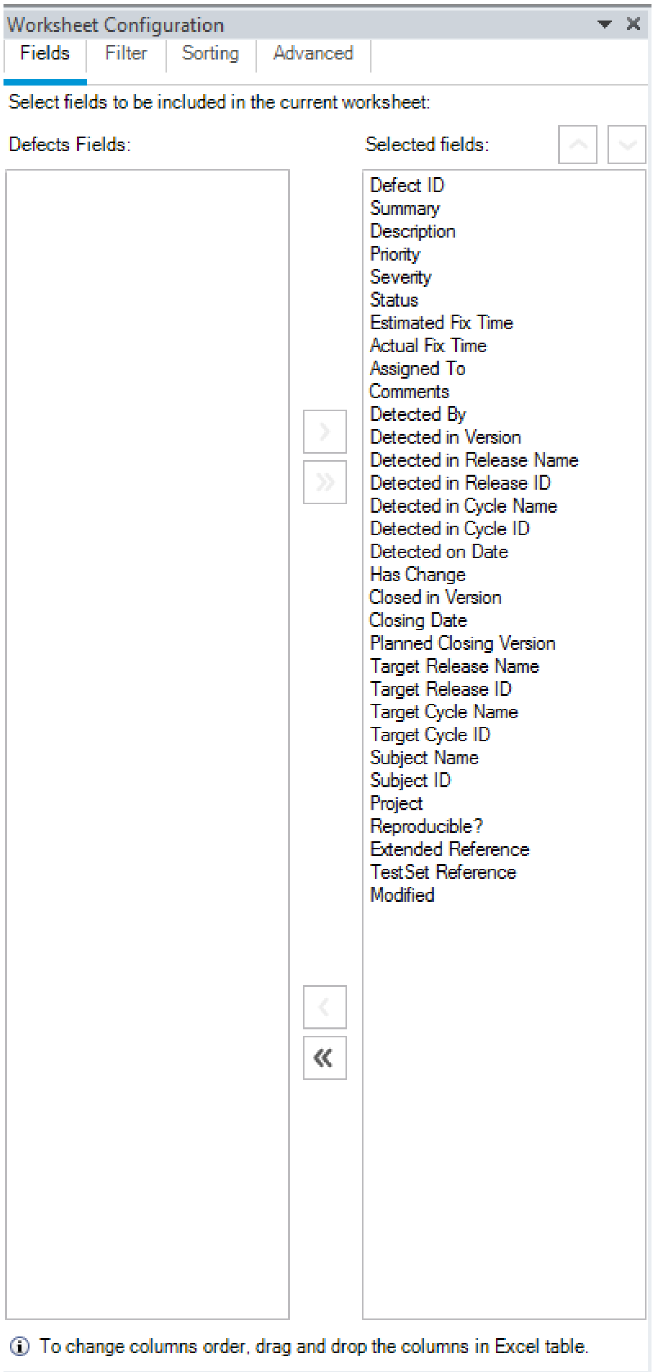
a. Click **Add**. The Add Worksheet dialog box opens.



Add Worksheet
Select one or more business views for the report worksheets:
Filter:
Business Views
Baselines (Baselines)
Components (Components)
Defects (Defects)
Defects Assigned to me (Defects_Assigned_to_me)
Defects With Linked Defects (Defects_With_Linked_Defects)
Defects With Linked Requirements (Defects_With_Linked_Req
Defects With Linked Tests (Defects_With_Linked_Tests)
Defects With No Linked_Req (Defects_With_No_Linked_Req)
Defects_history (Defects_history)
Release Cycles (Release_Cycles)
Releases (Releases)
ReleasesAndCycles (ReleasesAndCycles)
Requirements (Requirements)
Description:
OK
Cancel
Help

b. Select **Defects**.

c. Click **OK**. A new business view Excel worksheet is added and the Worksheet Configuration pane is opened.



Select fields to be included in the current worksheet:

Defects Fields:

Selected fields:

Defect ID
Summary
Description
Priority
Severity
Status
Estimated Fix Time
Actual Fix Time
Assigned To
Comments
Detected By
Detected in Version
Detected in Release Name
Detected in Release ID
Detected in Cycle Name
Detected in Cycle ID
Detected on Date
Has Change
Closed in Version
Closing Date
Planned Closing Version
Target Release Name
Target Release ID
Target Cycle Name
Target Cycle ID
Subject Name
Subject ID
Project
Reproducible?
Extended Reference
TestSet Reference
Modified

To change columns order, drag and drop the columns in Excel table.

5. Configure the business view report.

   a. In the **Filter** tab, select **Priority** in the **Field Name** column.

   b. In the associated **Criteria** column, click the arrow. The Select Filter Condition dialog box opens.

c. Click **4-Very High**, **Or**, and **5-Urgent**.

d. Click **OK**.

e. Select **Status** in the **Field Name** column.

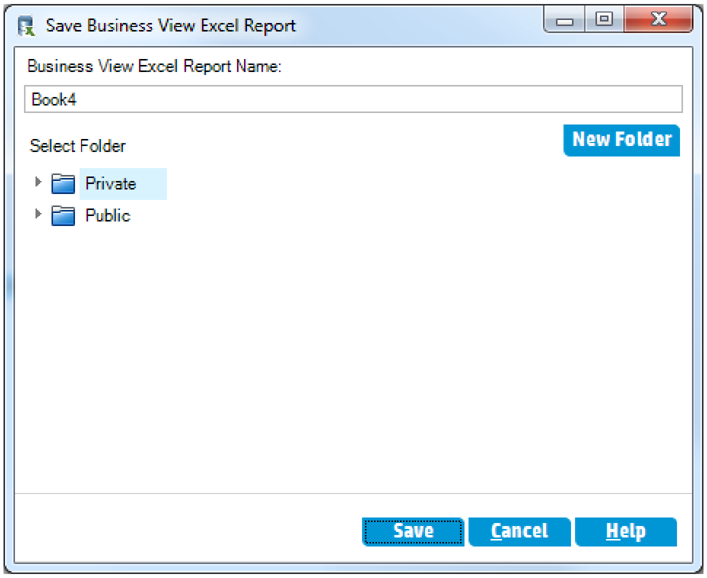
f. In the associated **Criteria** column, click the arrow. The Select Filter Condition dialog box opens.

g. Click **Not**, **Closed**, **And**, **Not**, and **Rejected**.

h. Click **OK**. The business view Excel report shows only those defects whose priority is **Very High** or **Urgent** and whose status is not **Closed** and not **Rejected**.

6. Save the new report in ALM.

a. Click **Save** and select **Save to ALM**. The Save Business View Excel Report dialog box opens.



Save Business View Excel Report

Business View Excel Report Name:

Book4

b. In the Select Folder box, browse to the **My_Analysis_Items** folder and select the High Priority Open Defects report. The Business View Excel Report Name box is automatically valued with High Priority Open Defects.

c. Click **Save**.

d. Click **Yes** in the Replace Confirm message. The report is saved to ALM.

7. Generate the report in ALM.
    a. Return to the Analysis View module in ALM.
    b. Click **Refresh**. The High Priority Open Defects report is selected with the Configuration tab open.
    c. Click **Generate**. The Download Excel File dialog box opens.
    d. Browse to the location on your client machine where you want to save the Excel report, enter High Priority Open Defects in the **File name** field, and click **Save**.

       The report opens in Excel with the latest data included.

Generating Dashboard Pages

Using the Dashboard module, you can arrange and view multiple graphs on a single dashboard page. You select the graphs to include in the dashboard page from the graphs in the analysis tree. You can arrange and resize the graphs on the page.

You create dashboard pages in either a public folder or a private folder. Dashboard pages in a **public** folder are accessible to all users. Dashboard pages in a **private** folder are accessible only to the user who created them.

In this exercise, you will create a dashboard page for the defect graphs in the public folder.

To generate a dashboard page:

1. Display the Dashboard View module.

   On the ALM sidebar, under **Dashboard**, select **Dashboard View**.
2. Add a page to the Public folder.
    a. In the tree, select the **Public** folder.
    b. Click the **New Page** button. The New Dashboard Page dialog box opens.

   c. In the **Dashboard Page Name** field, type Summary of Defects page.

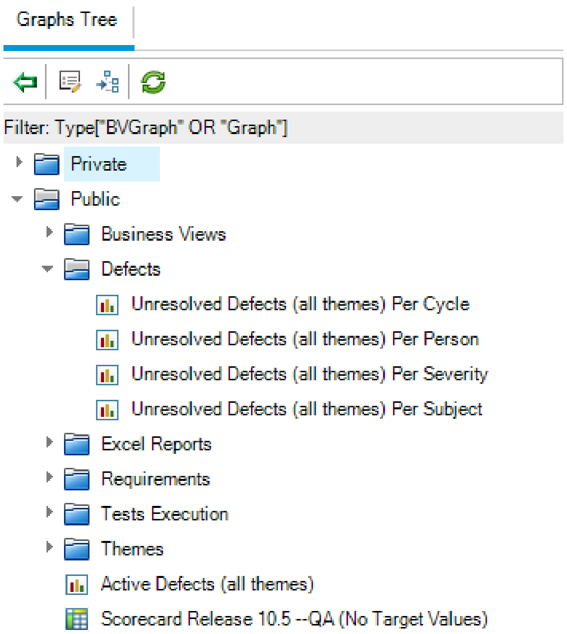
   d. Click **OK**.A dashboard page is added to the dashboard tree under the Public folder.

3. Select the graphs that you want to include in the dashboard page.

   a. Click the **Configuration** tab.

   b. In the Graphs Tree pane, expand the **Public** folder.

   c. Expand the **Defects** folder. The folder includes four graphs.



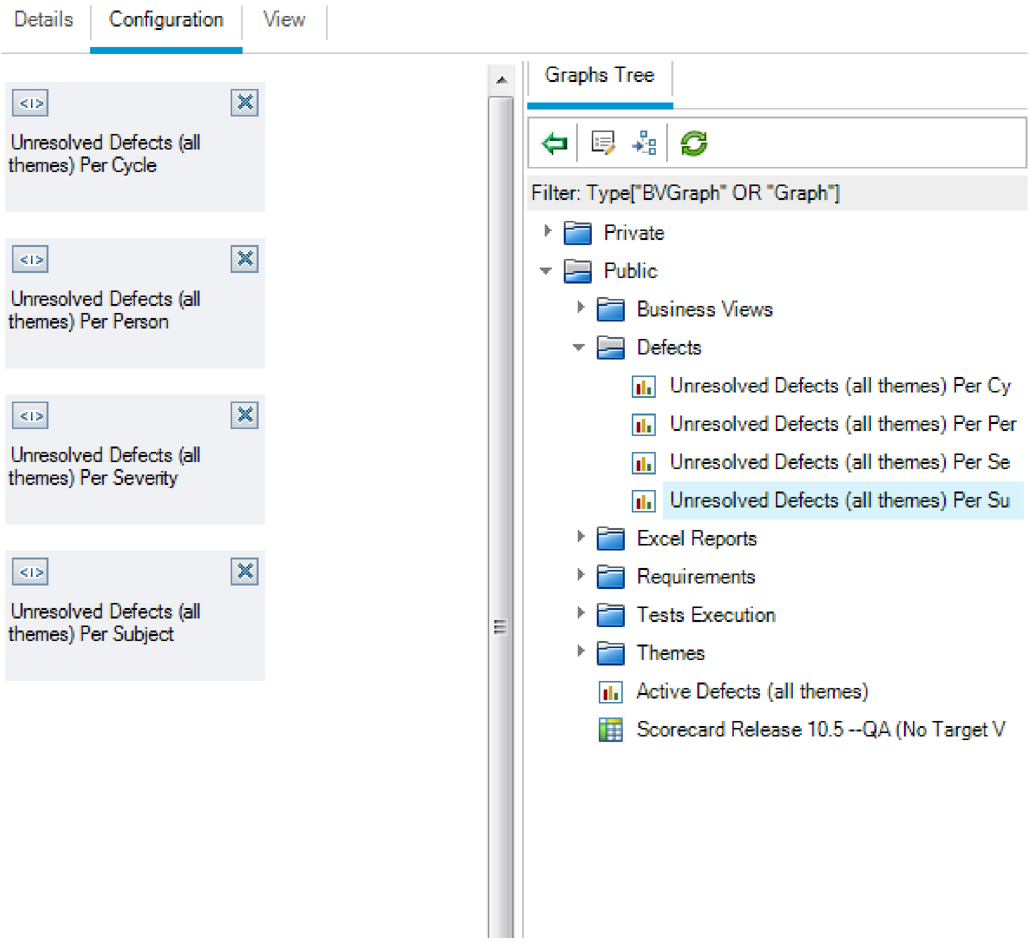
Graphs Tree

Filter: Type["BVGraph" OR "Graph"]

- Private
- Public
  - Business Views
  - Defects
    - Unresolved Defects (all themes) Per Cycle
    - Unresolved Defects (all themes) Per Person
    - Unresolved Defects (all themes) Per Severity
    - Unresolved Defects (all themes) Per Subject
  - Excel Reports
  - Requirements
  - Tests Execution
  - Themes
  - Active Defects (all themes)
  - Scorecard Release 10.5 --QA (No Target Values)

   d. Double-click the first graph. A placeholder for the graph is created in the Configuration tab displaying the graph’s title.

e. Add the other three graphs to the dashboard page.

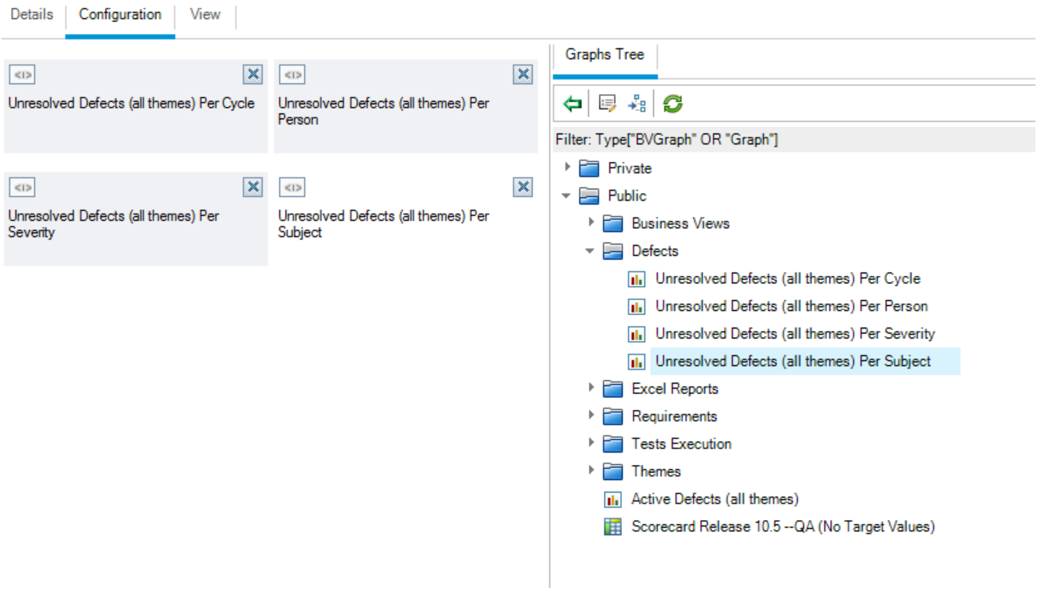


Details
Configuration
View
Unresolved Defects (all themes) Per Cycle
Unresolved Defects (all themes) Per Person
Unresolved Defects (all themes) Per Severity
Unresolved Defects (all themes) Per Subject
Graphs Tree
Filter: Type["BVGraph" OR "Graph"]
Private
Public
Business Views
Defects
Unresolved Defects (all themes) Per Cy
Unresolved Defects (all themes) Per Per
Unresolved Defects (all themes) Per Se
Unresolved Defects (all themes) Per Su
Excel Reports
Requirements
Tests Execution
Themes
Active Defects (all themes)
Scorecard Release 10.5 --QA (No Target V

4. Rearrange the dashboard page.

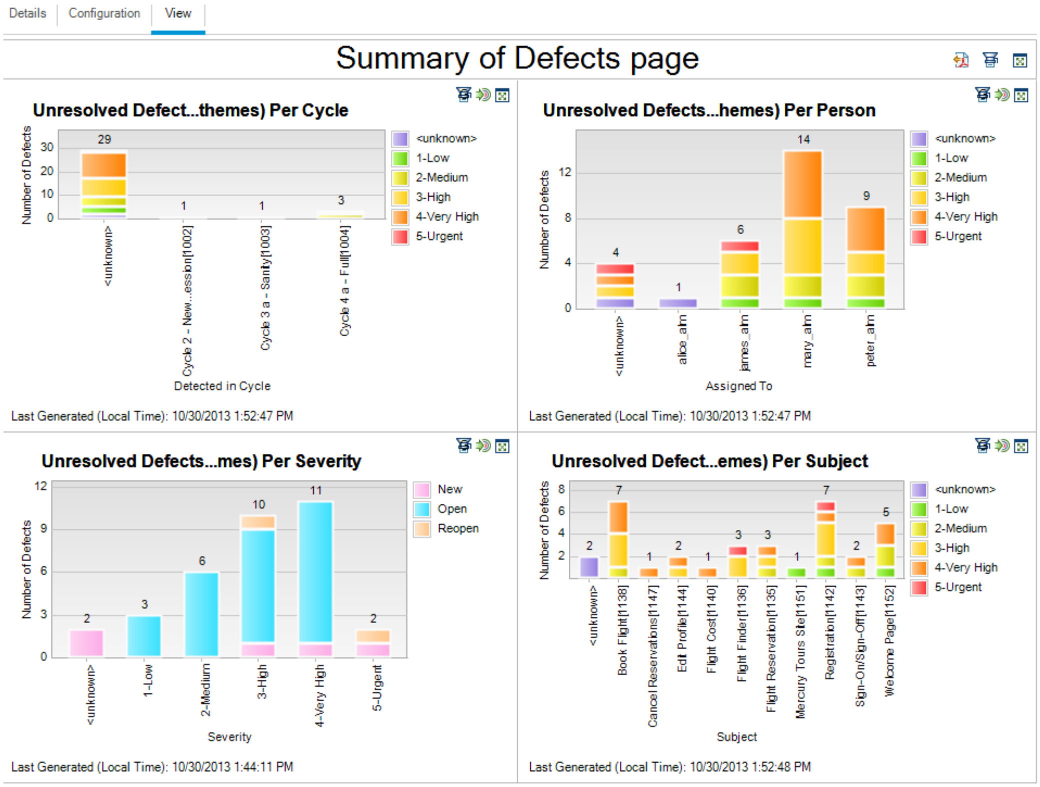
   a. Select the second placeholder and drag it upwards so that it is alongside the first placeholder.

b. Select the fourth placeholder and drag it upwards so that it is alongside the third placeholder.


Details
Configuration
View
Unresolved Defects (all themes) Per Cycle
Unresolved Defects (all themes) Per Person
Unresolved Defects (all themes) Per Severity
Unresolved Defects (all themes) Per Subject
Graphs Tree
Filter: Type["BVGraph" OR "Graph"]
Private
Public
Business Views
Defects
Unresolved Defects (all themes) Per Cycle
Unresolved Defects (all themes) Per Person
Unresolved Defects (all themes) Per Severity
Unresolved Defects (all themes) Per Subject
Excel Reports
Requirements
Tests Execution
Themes
Active Defects (all themes)
Scorecard Release 10.5 --QA (No Target Values)

5. View the dashboard page.

   a. Click the **View** tab. ALM generates and displays the graphs in the View tab.



The dashboard page displays the unresolved defects per cycle, per person, per severity, and per subject.

b. To view the dashboard page in full-screen mode, click the **View Page in Full Screen** button, located in the upper-right corner of the page.

c. To return to the standard view, click the **Close** button, located in the upper-right corner of the page.

Creating Libraries and Baselines

A library represents a set of entities in a project and the relationships between them. The entities in a library can include requirements, tests, test resources, and business components. A baseline is a snapshot of the library at a specific point in time. Baselines enable you to keep track of changes made to your project over time.

You create libraries in the Libraries module.

You can compare baselines at all stages of the application lifecycle management process. For example, you can compare two baselines in a library to review changes made to tests in the library over time. You can also compare a baseline to the current entities in the library.

In this lesson, you will create a library of tests and requirements. You will then compare two baselines in the library to review changes made to tests in the library over time.

In this lesson, you will learn about:

Creating Libraries

In this exercise, you will add a library of tests and requirements to ALM. To perform this exercise you must log in as alex_alm.

To create a library:

1. Make sure to log in to ALM_Demo as alex_alm.

   Open the ALM Login window. In the **Login Name** box, type alex_alm. Skip the **Password** box.

   For details, see "Start" on page 10.

2. Display the Libraries module.

   On the ALM sidebar, under **Management**, select **Libraries**.

3. Create a library folder.

   a. In the libraries tree, select the root **Libraries** folder. Click the **New Folder** button. The New Library Folder dialog box opens.

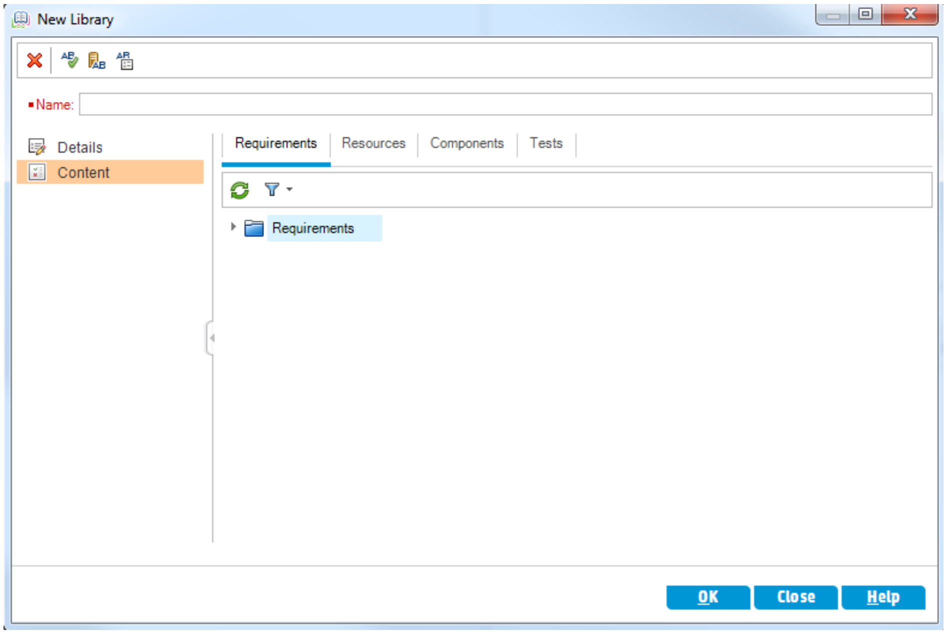
   b. In the **Library Folder Name** box, type Folder1.

   c. Click **OK**. The **Folder1** folder is added to the libraries tree.

   d. In the **Description** box in the right pane, type the following description for the library folder: This folder contains a library of tests and requirements.

4. Add a library to your library folder.

   a. Click the **Create Library** button. The New Library dialog box opens on the Content page.



b. In the **Name** box, type Library1.

c. In the **Requirements** tab, expand the **Requirements** root folder. Select the check box adjacent to the **Mercury Tours Application** folder, to include the folder in the library.

d. Click the **Tests** tab. Make sure the **Tests in selected folders** option is selected. Expand the **Subject** root folder, and select the **Mercury Tours Site** folder to include in the library.

e. Click **Details** on the sidebar. In the **Description** box, type This library includes tests and requirements.

f. Click **OK**. The new library is added to the libraries tree.

Creating Baselines

A baseline is a snapshot of your library at a specific point in time. You can use a baseline to mark any significant milestone in the application lifecycle management process. A baseline includes all the entities defined in the library, including requirements, tests, test resources, and business components. Baselines also include the relationships between the entities in the library, such as traceability and coverage. Baselines enable you to keep track of changes made to your project over time.

In the following exercise, you will create an initial baseline that will later be compared to another baseline to evaluate the impact of changes.

To create a baseline:

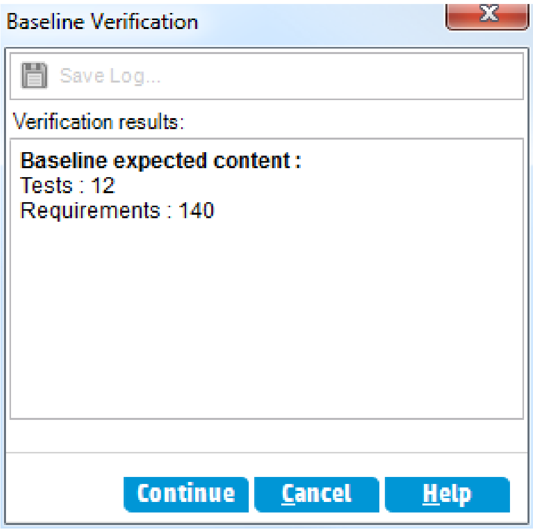
1. Make sure the Libraries module is displayed.

   If the Libraries module is not displayed, on the ALM sidebar, under **Management**, select **Libraries**.

2. Add a baseline to your library.

   a. In the libraries tree, select the **Library1** library.

   b. Click the **Create Baseline** button. The Baseline Verification dialog box opens, and a verification process begins.

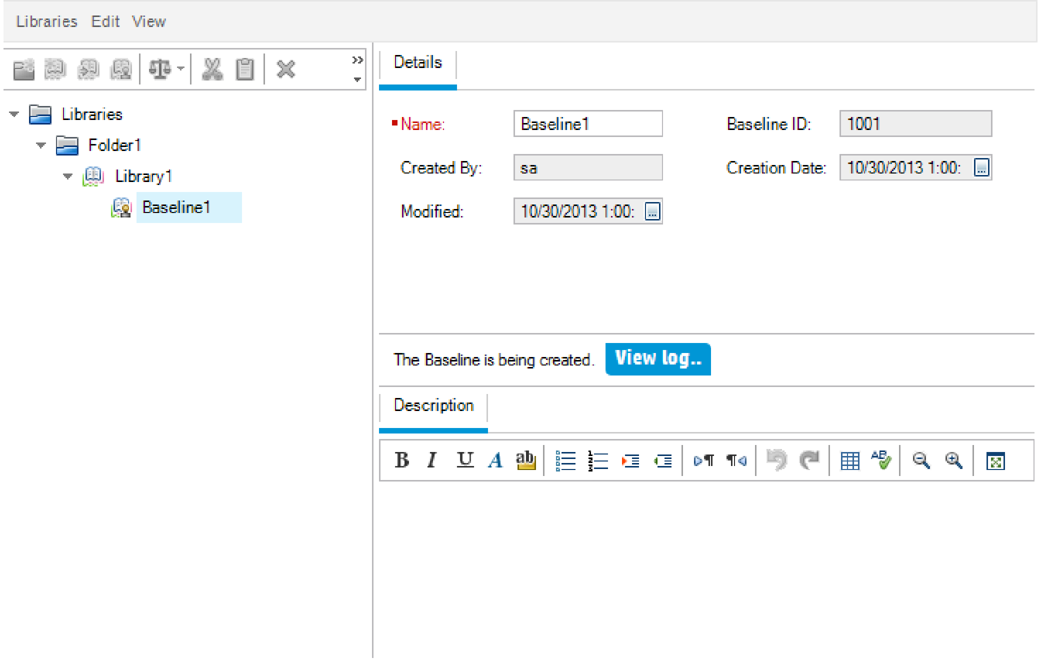
   Verification results:

   Baseline expected content :
   Tests : 12
   Requirements : 140

   The Verification results window displays the results of the library size verification. The process checks that the size of the library does not exceed the limit defined in Site Administration.

   c. Click **Continue**. The New Baseline dialog box opens.

   d. In the **Baseline Name** field, type Baseline1. Click **OK**. The baseline is added to the libraries tree, and the creation process begins.



Name: Baseline1

Baseline ID: 1001

Created By: sa

Creation Date: 10/30/2013 1:00:

Modified: 10/30/2013 1:00:

The Baseline is being created.

The baseline is created in a background process, and may take some time. You can continue working in ALM during the baseline creation process.

e. In the Details tab, click the **Description** pane and type Baseline of tests and requirements.

3. View the baseline log file.

   In the Details tab, click the **View Log** button. The Log: Create Baseline dialog box opens and displays the progress. Click **Close** to close the dialog box.

   The **View Log** button is no longer displayed.

Comparing Baselines

You can compare two baselines in a library. For example, you can compare baselines at different stages of development to assess the impact of changes made to requirements in your project. You can then update the relevant tests in your project accordingly.

You can also compare a baseline to the current entities in the library. For example, suppose you create a baseline at the start of a new release. Over time, changes are made to requirements in the library. To determine whether product development is proceeding as planned, you can compare requirements in the initial baseline with the current requirements in the library.

In the following exercise, you will add test coverage to a requirement and then create another baseline. You will then compare your two baselines to evaluate the impact of the changes.

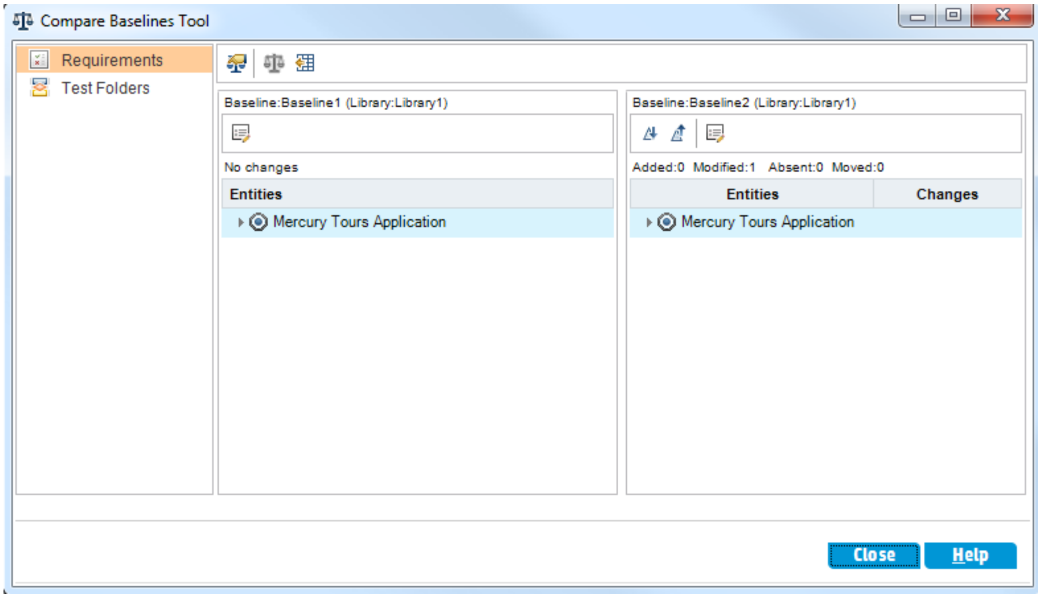
To compare baselines:

1. Modify a requirement.
   a. On the ALM sidebar, under **Requirements**, select **Requirements**.
   b. Select **View > Requirement Details**.
   c. In the requirements tree, under **Mercury Tours Application**, expand **Application Usability**. Select **Keyboard Support**.
   d. Click the **Test Coverage** tab. The Test Coverage tab displays coverage for this requirement.
   e. If the Test Plan Tree tab on the right is not displayed, click the **Select Tests** button.
   f. In the Test Plan Tree tab, expand the **Mercury Tours Site** and **HTML Pages** subject folders.
   g. Double-click the **HTML Page Source** test. The test is added to the coverage grid.
2. Create a new baseline.

   Repeat Steps 1 and 2 in "Creating Baselines" on page 149. Name your new baseline Baseline2.
3. Select a baseline with which to compare.
   a. In the libraries tree, select **Baseline1**. Click the **Compare To** button, and select **Select Baseline** to compare the baseline with another baseline. The Select Baseline dialog box opens.

b. Click the arrow and select **Baseline2** from the list. Click **OK**.

c. Click **OK** to close the Select Baseline dialog box. Click **Yes** to close the Warning dialog box. The Compare Baselines Tool dialog box opens.



Baseline:Baseline1 (Library:Library1)

No changes

Mercury Tours Application

Baseline:Baseline2 (Library:Library1)

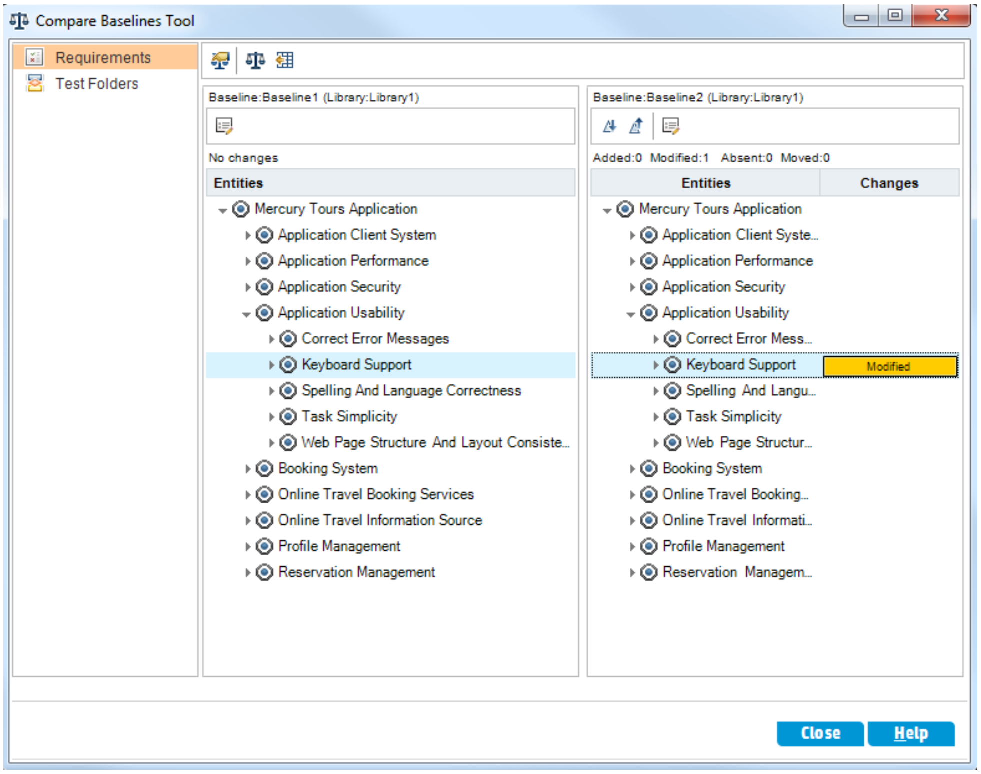
Added:0 Modified:1 Absent:0 Moved:0

Mercury Tours Application

The baselines are displayed in separate panes, with the more recently created baseline displayed in the right pane. In each pane, the library's entities are displayed in the same hierarchical tree structure as defined in the specific module.

4. View requirement changes between the baselines.

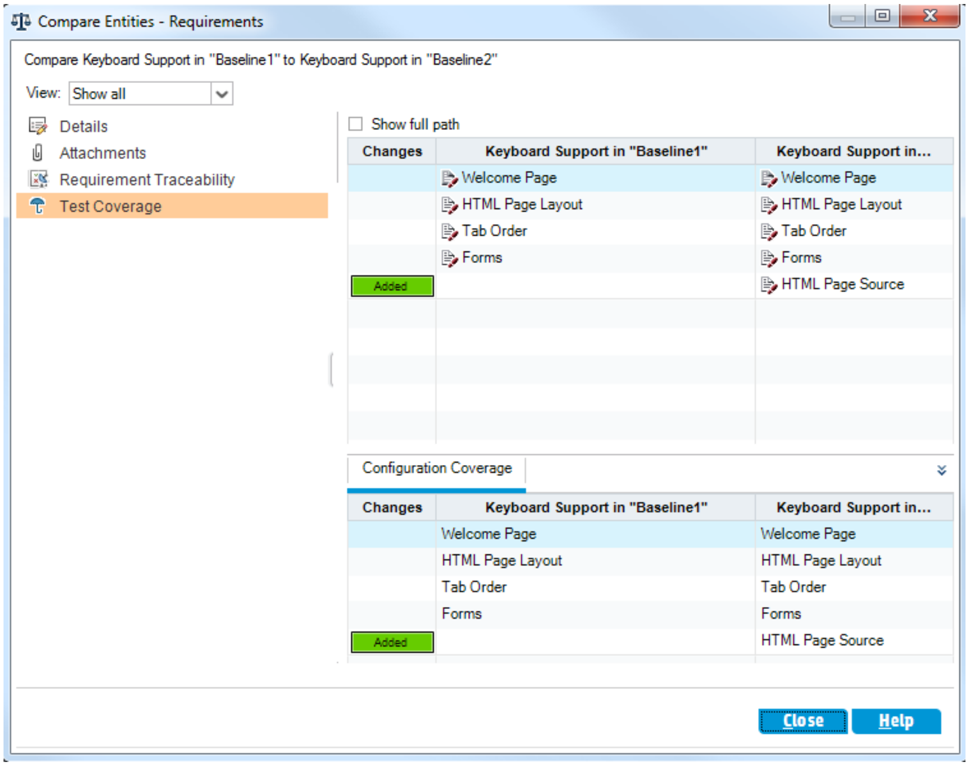
   a. Click the **Go To Next Change** button in the right pane to view the change.



Differences between the two baselines are indicated in the **Changes** column. The tool indicates that there is a difference in the **Keyboard Support** requirement between the baselines.

b. To compare the modified requirement between baselines, select **Keyboard Support** and click the **Compare Entities** button on the toolbar. The Compare Entities dialog box opens.

c. Click the **Test Coverage** button on the sidebar.



Compare Keyboard Support in "Baseline1" to Keyboard Support in "Baseline2"

View: Show all

□ Show full path

Changes	Keyboard Support in "Baseline1"	Keyboard Support in...
	Welcome Page	Welcome Page
	HTML Page Layout	HTML Page Layout
	Tab Order	Tab Order
	Forms	Forms
Added		HTML Page Source

Configuration Coverage

Changes	Keyboard Support in "Baseline1"	Keyboard Support in...
	Welcome Page	Welcome Page
	HTML Page Layout	HTML Page Layout
	Tab Order	Tab Order
	Forms	Forms
Added		HTML Page Source

The Test Coverage view displays details of the entity in each baseline.

d. Click **Close**.

Customizing Projects

In the previous lessons, you learned how to use ALM to help you manage all phases of the application lifecycle management process, including specifying releases and cycles, specifying requirements, planning tests, running tests, and tracking defects.

In this lesson, you will learn how to customize your ALM project to meet the needs of your team. You can control access to a project by defining the users who can access the project and by specifying the types of tasks each user can perform. When new members are added to your team, you assign them to the projects that they will be using, and specify the tasks that they can perform.

You can also customize your ALM project by modifying system fields or by adding user-defined fields. **System fields** are ALM default fields. You cannot add or delete system fields, you can only modify them. **User fields** are fields that you can define. You can add, modify, and delete user-defined fields.

Fields can be associated with system and user-defined lists. A list contains the values that the user can enter in a field. For example, if you are running tests on two different database servers, you can add a **Database** field to your project. You can then create a selection list containing the values **Oracle** and **Microsoft SQL**, and associate the list with the **Database** field.

In this lesson, you will learn about:

Starting Project Customization

You customize your ALM projects using the Project Customization window. In this exercise, you will log in to the Project Customization window with project administrator privileges.

To start project customization:

1. Open the ALM Login window.

   Make sure that the ALM Login window is open. For details, see "Start" on page 10.

2. Type a user name with project administrator privileges and authenticate.

   a. In the **Login Name** box, type alex_alm.

   b. Skip the **Password** box. A password was not assigned to this user name.

   c. Click the **Authenticate** button. ALM verifies your user name and password and determines which domains and projects you can access.

3. Log in to the project.

   a. In the **Domain** list, select **Default**.

   b. In the **Project** list, select **ALM_Demo**.

   c. Click the **Login** button.

      The ALM main window opens and displays the module in which you were last working.

4. Open the Project Customization window.

   a. On the OpenText Application Quality Management masthead, click ⚙⌄ and select **Customize**. The Project Customization window opens.

      By default, the Project Customization window contains the following links:

Option	Description
User Properties	Enables you to change your user properties. For example, you can change your email address. You can also change your password.
Project Users	Enables you to add and remove users from an ALM project. You can also assign users to user groups to restrict user access privileges.
Groups and Permissions	Enables you to assign privileges to user groups by specifying permission settings.
Module Access	Enables you to control the modules that each user group can access. By preventing users from accessing unnecessary modules, you can better utilize your ALM licenses.
Project Entities	Enables you to modify the behavior of ALM system fields or define user-defined fields that are unique to your project. For example, if you are running tests on several builds of an application, you can add a **Detected in Build** user-defined field to the New Defect dialog box. You can then associate it with a selection list containing the values for this field.
Requirement Types	Enables you to customize the definitions for requirement types.
Risk-Based Quality Management	Enables you to customize settings for risk-based quality management.
Project Lists	Enables you to add customized lists to a project. A list contains values that the user can enter in system or user-defined fields. For example, for the **Detected in Build** field, you can create a selection list containing the values **Build1**, **Build2**, and **Build3**.
Automail	Enables you to set up automatic mail notification rules to inform users via email each time changes are made to specified defects.

Option	Description
Alert Rules	Enables you to activate alert rules for your project. This instructs ALM to create alerts and send emails when changes occur in the project.
Workflow	Enables you to generate scripts that perform commonly needed customizations on dialog box fields in the Defects module. In addition, you can write scripts to customize dialog boxes in other modules, and control the actions that users can perform.
Project Planning and Tracking	Enables you to create and customize the project planning and tracking (PPT) KPIs.
Project Report Templates	Enables you to create and customize report templates that project users can assign to template based reports.
Business Process Testing	Enables you to configure Business Process Testing and Business Process Testing Enterprise Edition.
Business Views	Enables you to create business views that can be used as a basis for creating reports in the Analysis View module.
Sprinter	Enables you to configure settings for working with Sprinter for manual testing in ALM.

Adding a New Project User

You can control access to an ALM project by defining the users who can log in to the project, and by specifying the types of tasks each user may perform.

For each project, you select project users from the ALM site users list. This list is created in Site Administration.

From Project Customization, you add users to a project and assign them to user groups. Each user group has access to certain ALM tasks.

In this exercise, you will add a new project user to the ALM_Demo project.

> **Note:** For the purpose of the exercise, you will first remove a user from the project, and then add the user to the project again.

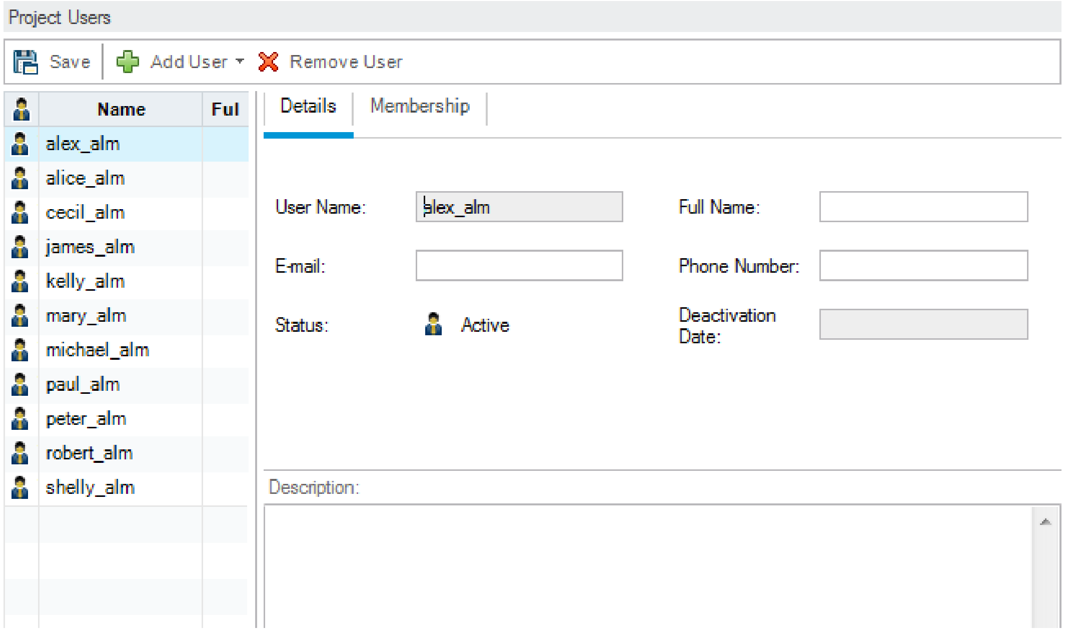
To add a new project user:

1. Make sure that you are logged in to ALM as a project administrator.

   For details on how to open the Project Customization window, see "Starting Project Customization" on page 157.

2. Open the Project Users page.

   In the Project Customization window, click the **Project Users** link. The Project Users page opens and displays a list of users that have been assigned to the project.

Name	Ful
alex_alm	
alice_alm	
cecil_alm	
james_alm	
kelly_alm	
mary_alm	
michael_alm	
paul_alm	
peter_alm	
robert_alm	
shelly_alm	

   User Name: alex_alm
   Full Name:
   E-mail:
   Phone Number:
   Status: Active
   Deactivation Date:

3. Remove a user.

   In the Project Users list, select **cecil_alm**, and click **Remove User**. Click **Yes** to confirm.

4. Add a new user name.

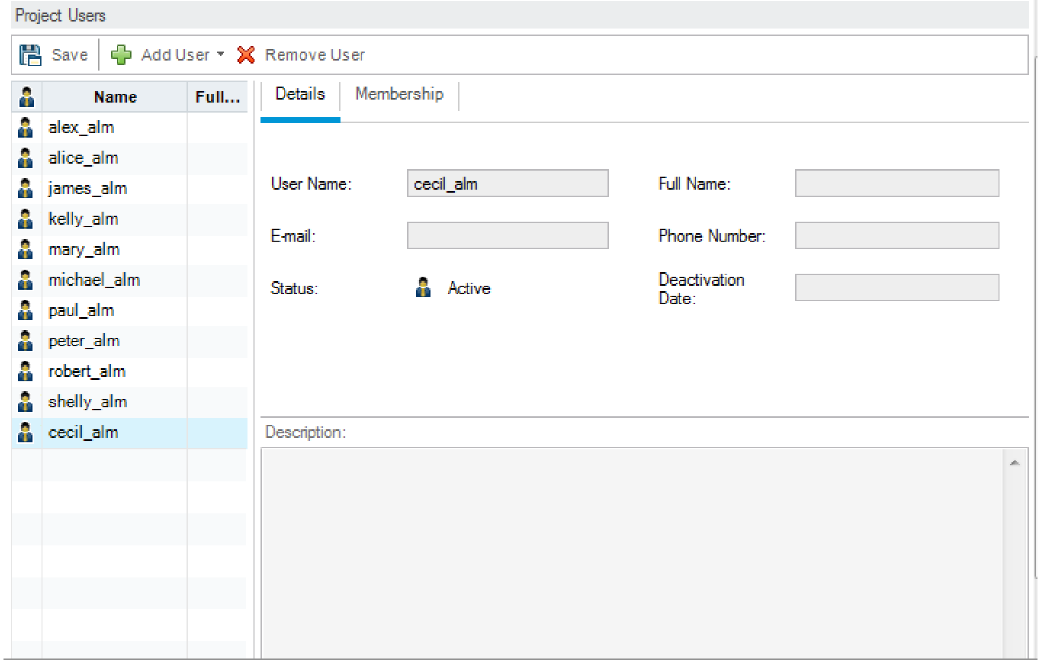
a. Click the **Add User** down arrow.

   You can add an existing user from the list of site users by typing the user's name or by selecting the user from the list of site users. You can also create a new user and add the new user to the project.

b. Select **Add User by Name**. The Add User dialog box opens.

c. In the **User Name** box, type **cecil_alm** and click **OK**.

   The new user is added to the Project Users list and the user properties are displayed in the Details tab. User personal settings are defined in Site Administration.



User Name: cecil_alm

Full Name:

E-mail:

Phone Number:

Status: Active

Deactivation Date:

Description:

Assigning a User to a User Group

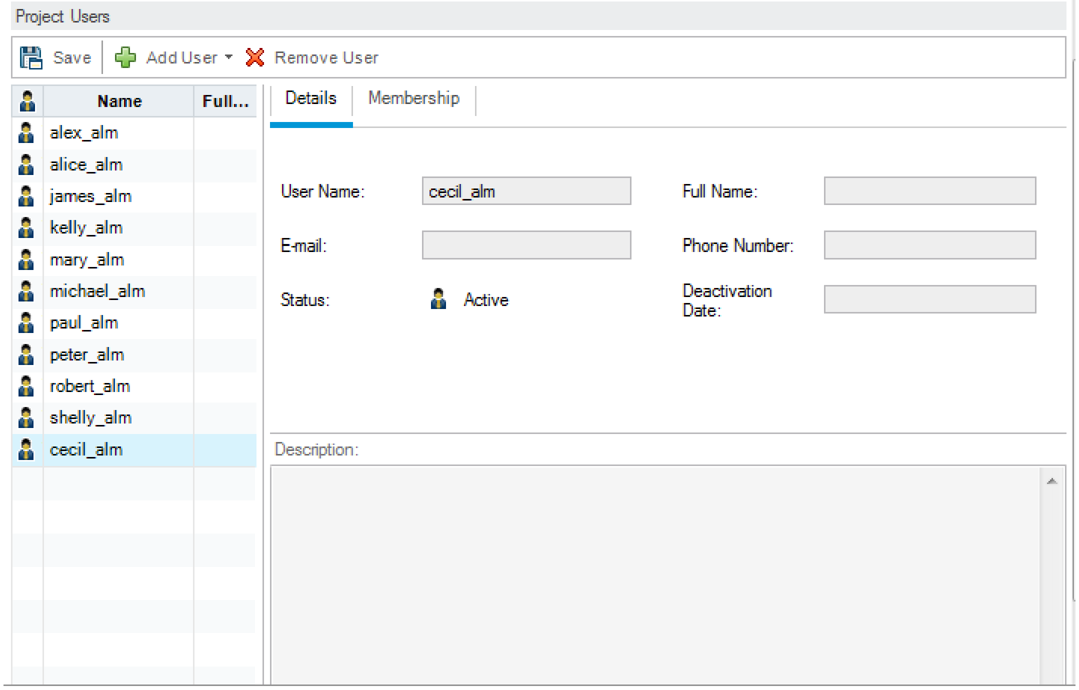
To enable users to do their job, and to protect a project from unauthorized access, ALM enables you to assign each user to a specific user group. Each group has access to certain ALM tasks. You can use the predefined user groups with their default permissions or you can customize your own user groups with unique sets of permissions.

In this exercise, you will assign the new user cecil_alm to the QATester user group.

To assign a user to a user group:

1. Make sure that the Project Users page is displayed.

   If the Project Users page is not already open, click the **Project Users** link in the Project Customization window.



Name	Full...
alex_alm	
alice_alm	
james_alm	
kelly_alm	
mary_alm	
michael_alm	
paul_alm	
peter_alm	
robert_alm	
shelly_alm	
cecil_alm	

User Name: cecil_alm

Full Name:

E-mail:

Phone Number:

Status: Active

Deactivation Date:

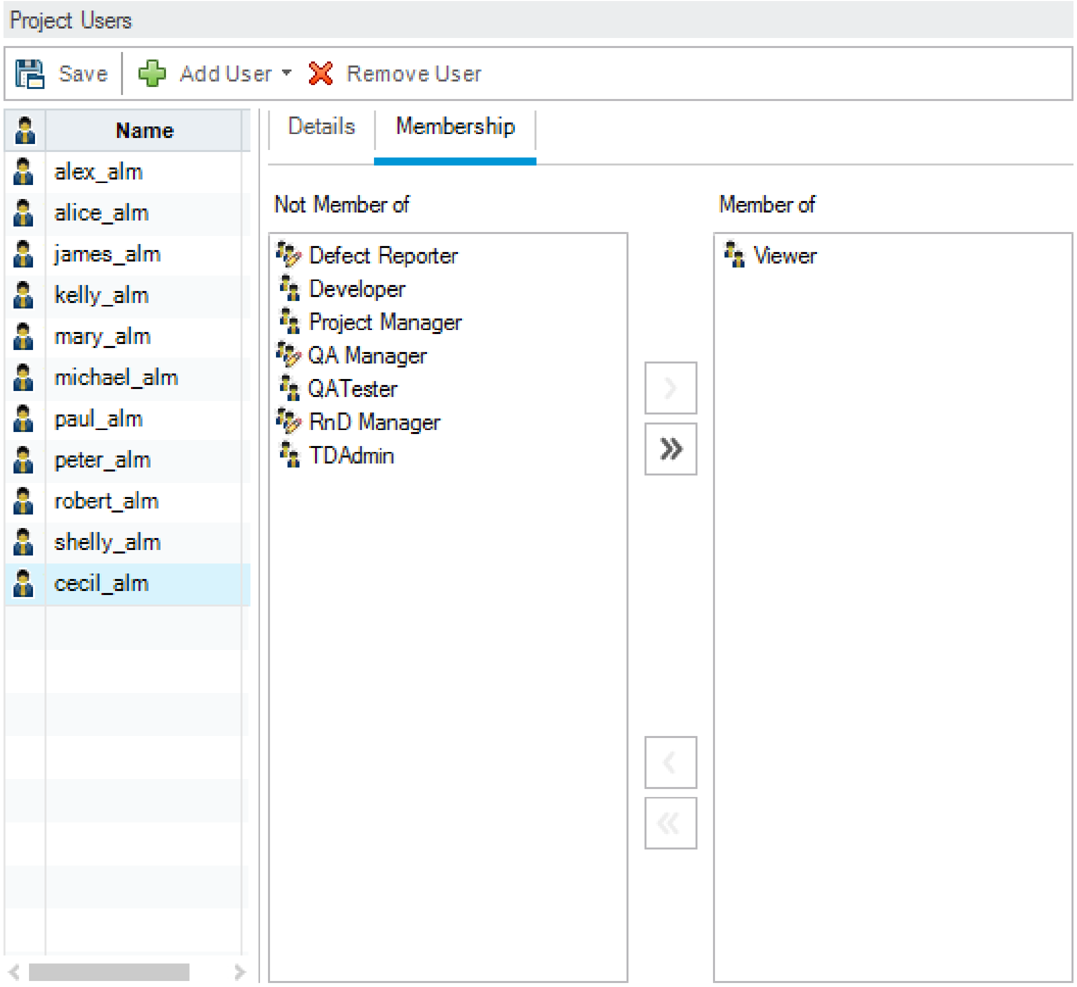
Description:

2. Select cecil_alm from the Project Users list.

   In the **Project Users** list, select **cecil_alm**.

3. Display user’s membership in user groups.

   Click the **Membership** tab. The user groups to which cecil_alm belongs and does not belong are displayed.



4. Assign cecil_alm to the QATester group.

   Under **Not Member of**, select **QATester** and click the right arrow button to move the group to **Member of**.

5. Remove cecil_alm from the Viewer group.

   a. Under **Member of**, select **Viewer** and click the left arrow button to move the group to **Not Member of**.

   b. Click **Save** to save the changes to the Project Users page. Click **OK**.

Defining a User-Defined Field

You can define user-defined fields that are unique to your project, or modify the behavior of ALM system fields.

The fields are stored in ALM project entities. For example, the Defect entity contains data entered in the Defects module.

In the following exercise, you will add the **Database** user-defined field to the **Defect** entity. This field indicates the server database in use when testing an application.

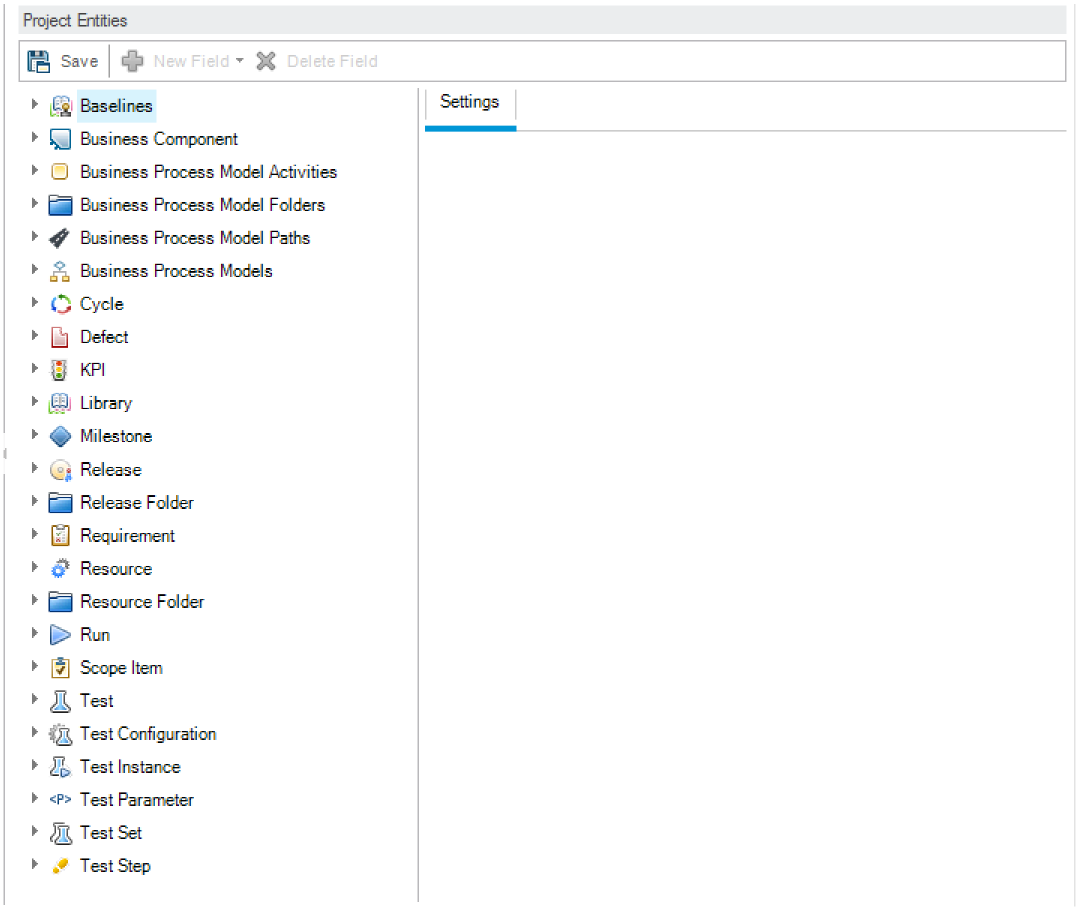
To add a user-defined field:

1. Make sure that the Project Customization window is displayed.

   For details on how to open the Project Customization window, see "Starting Project Customization" on page 157.

2. Open the Project Entities page.

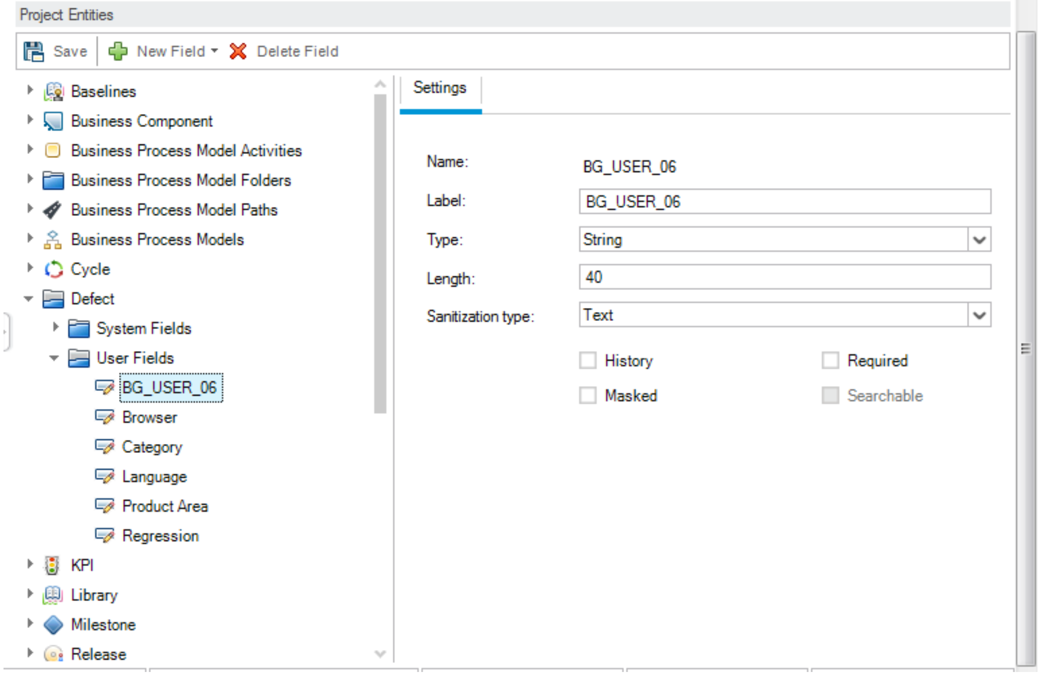
   In the Project Customization window, click the **Project Entities** link. The Project Entities page opens.



3. Add a new user-defined field to the Defect entity.

   a. Under **Project Entities**, expand **Defect**.

   b. Click the **User Fields** folder and click the **New Field** button. A new field is added under the User Fields folder.



Name: BG_USER_06

Label: BG_USER_06

Type: String

Length: 40

Sanitization type: Text

□ History □ Required

□ Masked □ Searchable

BG_USER_nn indicates a user-defined field under the Defect entity.

4. Rename the default field name.

   a. In the **Label** box, instead of the default name, type Database.

   b. Click **Save**.

   c. Click **OK**.

Creating a Project List

You can associate fields with system and user-defined lists. A list contains values that the user can enter in a field.

In the previous exercise you added the Database field. In the following exercise you will create a list and assign it to the Database field. You will then open the New Defect dialog box to view the new field.

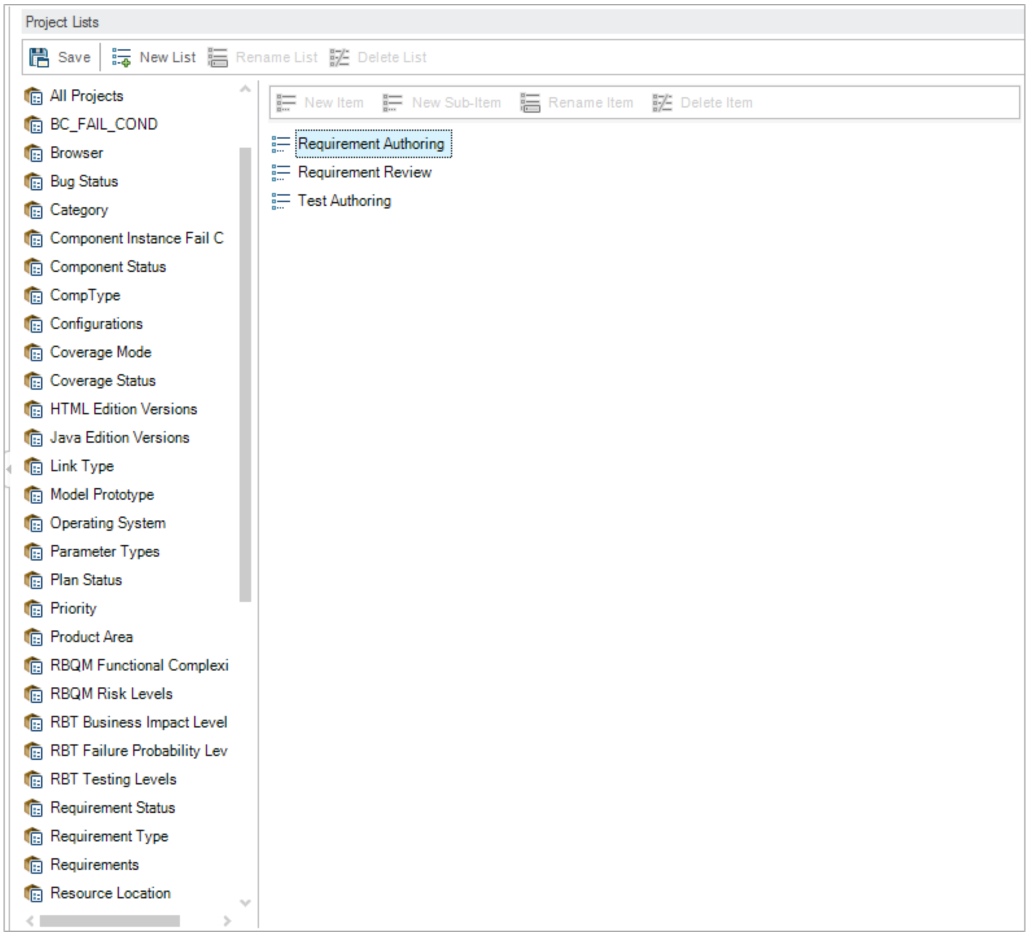
To create a project list:

1. Make sure that the Project Customization window is displayed.

   For details on how to open the Project Customization window, see "Starting Project Customization" on page 157.

2. Open the Project Lists page.

   In the Project Customization window, click the **Project Lists** link. The Project Lists page opens.

   All Projects
   BC_FAIL_COND
   Browser
   Bug Status
   Category
   Component Instance Fail C
   Component Status
   CompType
   Configurations
   Coverage Mode
   Coverage Status
   HTML Edition Versions
   Java Edition Versions
   Link Type
   Model Prototype
   Operating System
   Parameter Types
   Plan Status
   Priority
   Product Area
   RBQM Functional Complexi
   RBQM Risk Levels
   RBT Business Impact Level
   RBT Failure Probability Lev
   RBT Testing Levels
   Requirement Status
   Requirement Type
   Requirements
   Resource Location

   Requirement Authoring
   Requirement Review
   Test Authoring

3. Create a new list.

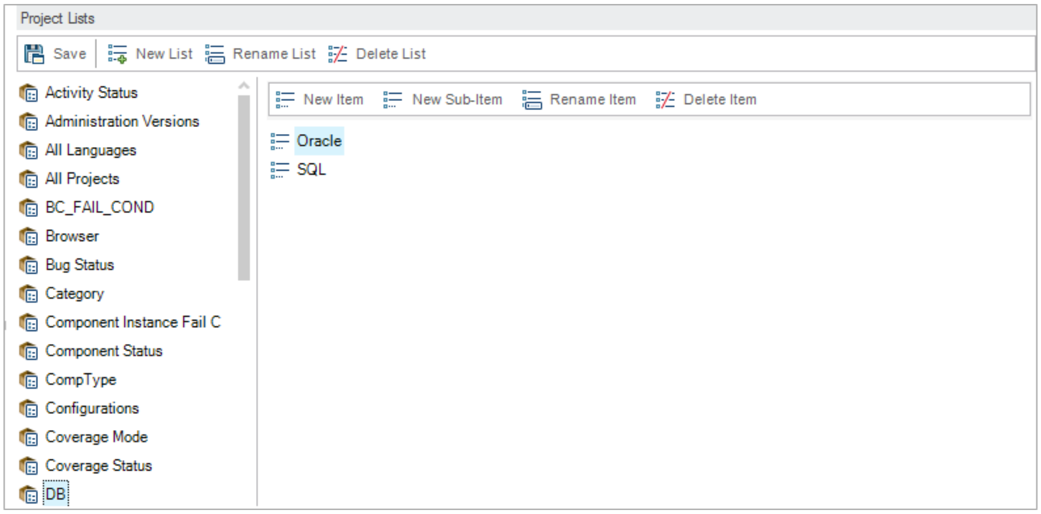
   a. Click **New List**. The New List dialog box opens.

   b. In the **List Name** box, type DB. Click **OK** to close the New List dialog box.

4. Add Items to your list.

a. Click **New Item**. The New Item dialog box opens. Type **Oracle** and click **OK**.

b. Repeat the same procedure and add **MS SQL** to the DB list.



c. Click **Save** to save the changes to the Project Lists page. Click **OK**.

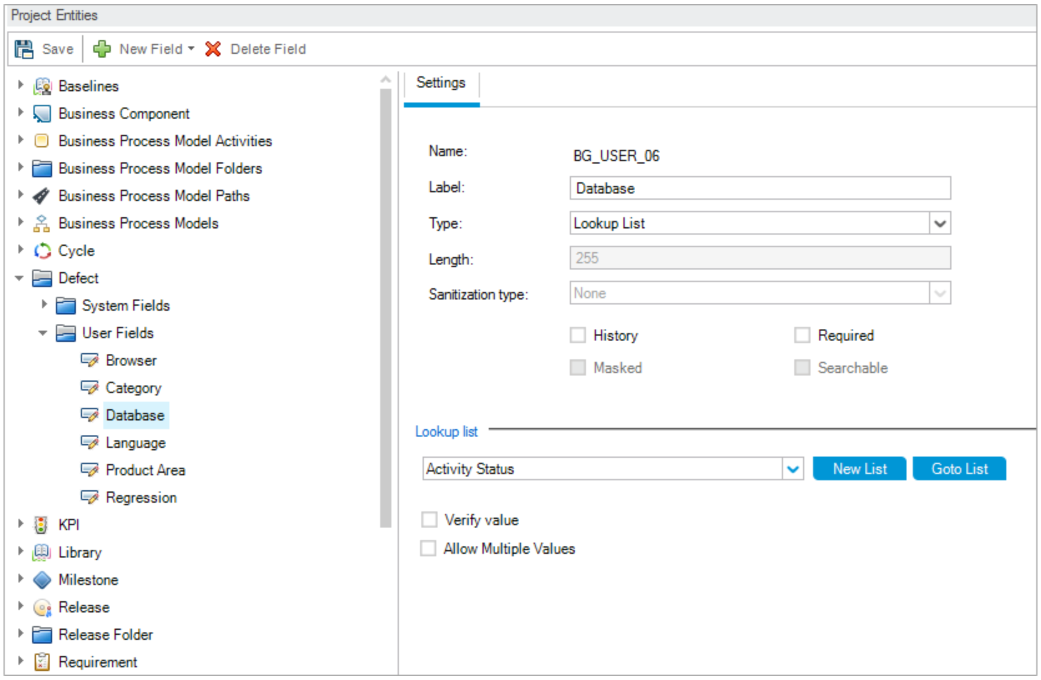
5. Assign the list to the Database field.

   a. In the Project Customization window, click the **Project Entities** link. The Project Entities page opens.

   b. Under **Project Entities**, expand **Defect**.

   c. Expand the **User Fields** folder and select **Database**.

   d. Under **Settings**, in the **Type** list, select **Lookup List** to set the field type as a drop-down list. The **Lookup list** section is displayed below the field settings.



Name: BG_USER_06

Label: Database

Type: Lookup List

Length: 255

Sanitization type: None

□ History □ Required

□ Masked □ Searchable

Lookup list: Activity Status

□ Verify value

□ Allow Multiple Values

   e. Under **Lookup List**, select the **DB** list.

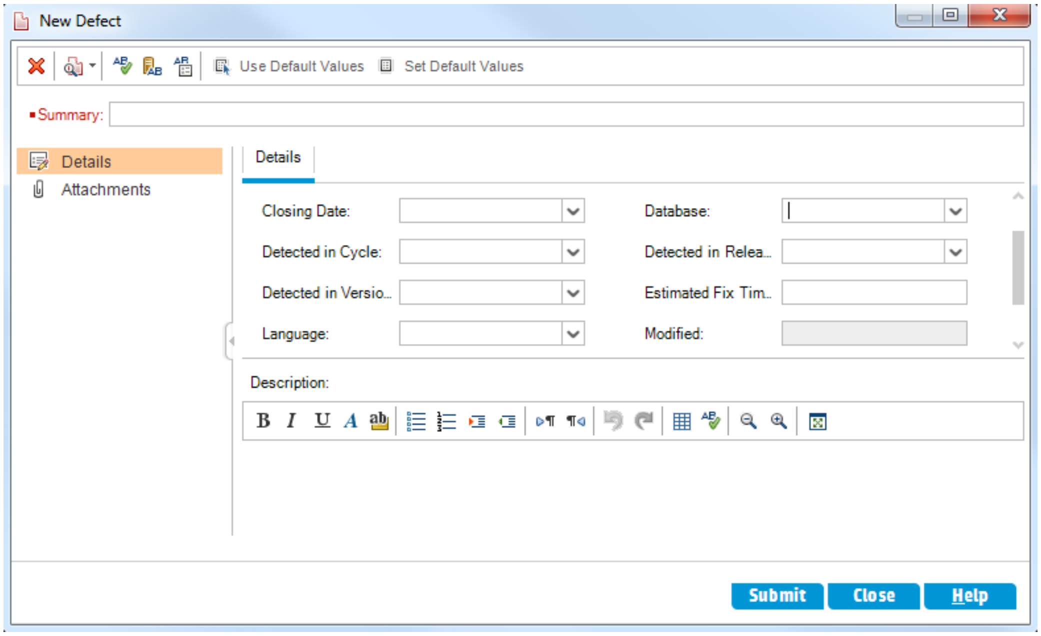
   f. Click **Save** to save the changes to the Project Entities page. Click **OK**.

6. View the new user-defined field in the New Defect dialog box.

   a. Click the **Return** button located on the upper-right corner of the window.

   b. In the Customization Changes dialog box, select **Major Change**, and click **OK**. The Project Customization window closes, and you return to your ALM project.

   c. In the Defects module, click the **New Defect** button. The New Defect dialog box opens.



The **Database** field is displayed in the New Defect dialog box. You may need to drag the scroll bar down to display the field.

d. Click the down arrow and view the database types you defined. Click **Close**.

Creating Business Views

Business views are a semantic data layer that can be used as a basis for the various ALM reporting tools. Business views are based on project entities, and ensure that only information that is relevant to a business consumer is contained in the report.

In the following exercise you will create a business view for creating graph reports.

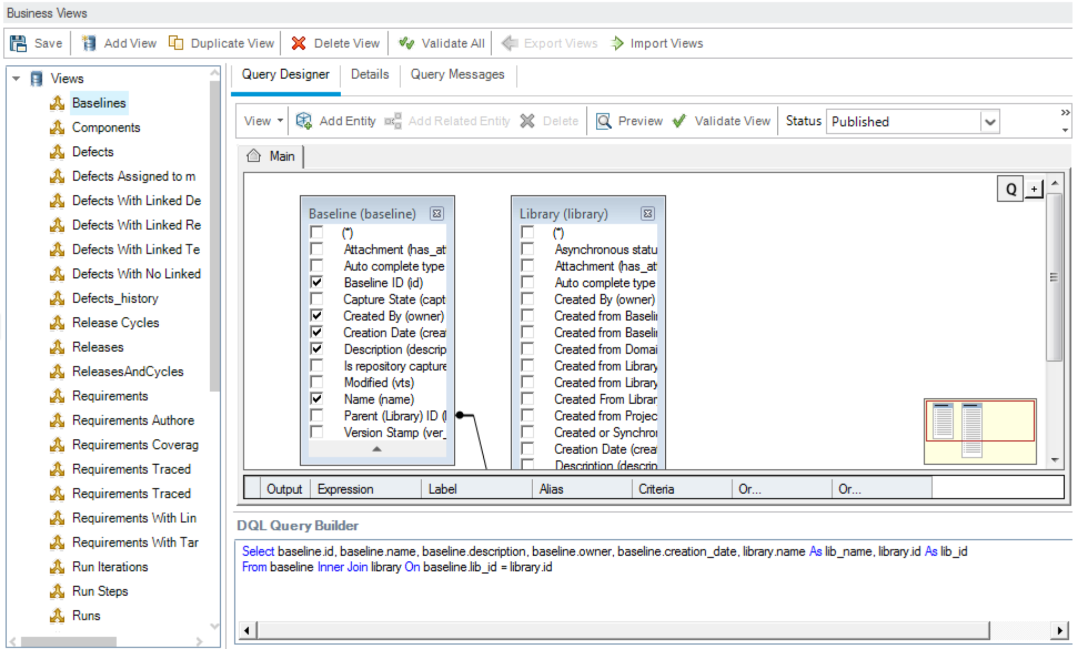
To create a business view:

1. Make sure that the Project Customization window is displayed.

   For details on how to open the Project Customization window, see "Starting Project Customization" on page 157.

2. Open the Business Views page.

   In the Project Customization window, click the **Business Views** link. The Business Views page opens.



Select baseline.id, baseline.name, baseline.description, baseline.owner, baseline.creation_date, library.name As lib_name, library.id As lib_id
From baseline Inner Join library On baseline.lib_id = library.id

3. Create a new business view.

   a. Click **Add View**. The New View dialog box opens.

   b. In the **Label** box, type Mercury Tour Defects. In the **Technical Name** box, type MT_Defects. Click **OK** to close the New View dialog box.

4. Add project entities to your business view.

   a. Click **Add Entity** in the Query Designer tab. The Model tree opens in the right pane. The Model tree displays all project entities and fields in the current project.

   b. Select the **Defect** entity and click the **Add** arrow to move it to the Main pane.

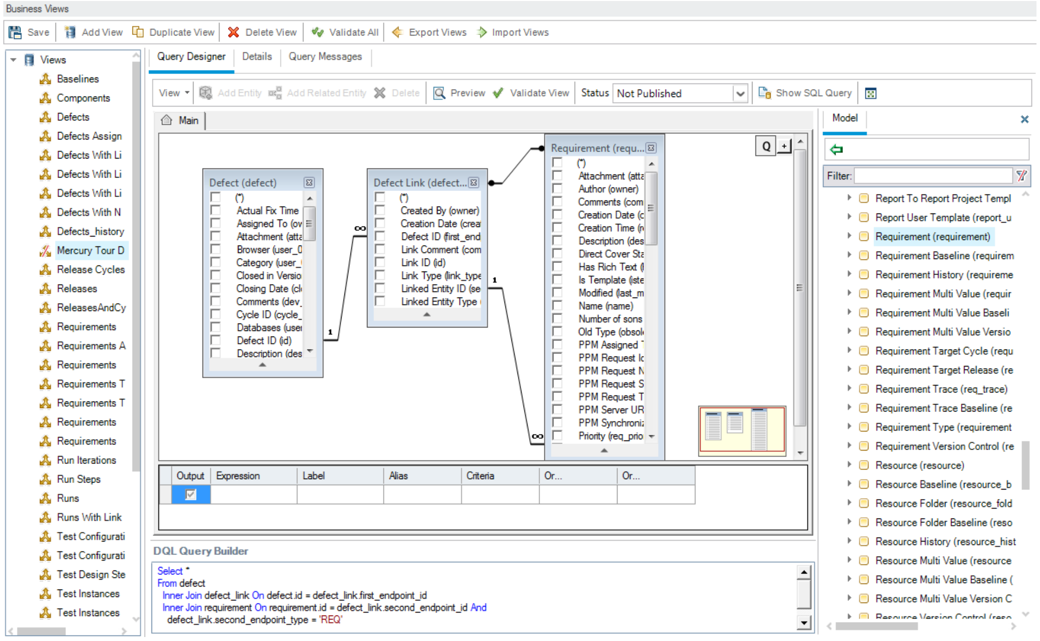
5. Define relationships between the project entities.

   a. In the Main pane, select the **Defect** entity.

   b. Click **Add Related Entity**. The Add Related Entity dialog box opens.

c. In **Target Entity**, click the arrow and select **Requirement**. The **Relation Name** field is automatically valued with **Linked Requirements**.

d. Click **OK**.



Status: Not Published

Output	Expression	Label	Alias	Criteria	Or...	Or...
☑						

```
Select *
From defect
  Inner Join defect_link On defect.id = defect_link.first_endpoint_id
  Inner Join requirement On requirement.id = defect_link.second_endpoint_id And
   defect_link.second_endpoint_type = 'REQ'
```

6. Preview the new business view.

   a. Click **Validate**. Any validation warning or error messages are shown in the bottom pane.

   b. Click **Preview**. The Query Results are shown in the bottom pane.

   c. Change the **Status** to Published.

   d. Click **Save**.