HP Service Test Management

Software Version: 11.00

User's Guide

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Java is a registered trademark of Oracle and/or its affiliates.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest

- Submit and track support cases and enhancement requests

- Download software patches

- Manage support contracts

- Look up HP support contacts

- Review information about available services

- Enter into discussions with other software customers

- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Table of Contents

Reference..**71**

Web Service Specific User Interfaces 71

Chapter 4: Managing Application Components **89**

Concepts...**90**

Managing Application Components Overview 90

Tasks **93**

How to Manage Components 93

Chapter 5: Groups..**97**

Concepts...**98**

Groups Overview...98

Tasks **101**

How to Manage Component Groups..101

Reference..**103**

New Groups User Interface 103

Chapter 6: Generating Requirements and Tests **105**

Concepts...**106**

Generating Requirements and Tests Overview...............................106

Tasks **110**

How to Generate Requirements and Tests..110

Reference..**112**

Generate Requirements and Tests User Interface 112

Chapter 7: Determining Test Coverage...................................**119**

Concepts...**120**

Coverage Overview 120
Rule Violations 120
Aspect Coverage 121
Requirements Coverage..122
Operations Coverage 123
Tests Coverage 124
Defects Coverage 126

Welcome to This Guide

Welcome to HP Service Test Management, a complete solution for managing the process of testing application components and their changes in service-oriented architecture (SOA) and other GUI-less systems. Service Test Management integrates with HP Application Lifecycle Management (ALM) to provide a Web-based solution for testing the quality and performance of all types of application components throughout the entire application development life cycle.

Note: This guide explains how to manage tests in conjunction with Application Lifecycle Management. For additional information on using ALM, refer to the *HP Application Lifecycle Management User Guide*.

Documentation Library

The Documentation Library is a set of guides that describe Service Test Management, the extension for ALM that allows you to manage application components, such as Web services.

The Documentation Library provides access to all guides available for Service Test Management, such as User guides, Helps, reference files, and links to printer-friendly (PDF format) documentation. It contains various navigation options to help you find the information you need.

To access the Documentation Library Home page for Service Test Management:

➤ Click the **Home** ⌂ icon, located at the top left corner of every content page.

➤ Click the **Home** entry on the **Contents** tab of the Documentation Library **Navigation** pane.

To access the Documentation Library Home page for the general HP ALM Help:

➤ Select **Help** > **Documentation Library**.

Documentation Library Guides

There are two documentation libraries that are available to you: A general one for Application Lifecycle Management (ALM) and another library specific to Service Test Management.

The following guides are available online, in PDF format, or both. PDFs can be read and printed using Adobe Reader which can be downloaded from the Adobe Web site *(*http://www.adobe.com*)*.

Service Test Management Guides

Guide	Description
HP Service Test Management User Guide	Explains how to use Service Test Management to facilitate the complete application delivery life cycle for application components. (Available online and in PDF format.)
HP ALM Installation Guide	Explains how to install Service Test Management on a server machine as an extension to ALM. (Available in PDF format only.)
Readme	Provides last-minute news and information about Service Test Management.

Application Lifecycle Management Guides

Guide	Description
HP ALM User Guide	Explains how to use HP ALM to organize and execute all phases of the application life cycle management process. It describes how to specify releases, define requirements, plan tests, run tests, and track defects.
HP ALM Administrator Guide	Explains how to create and maintain projects using Site Administration, and how to customize projects using Project Customization.
HP ALM Tutorial	A self-paced guide teaching you how to use HP ALM to manage the application life cycle management process.
HP ALM Installation Guide	Describes the installation and configuration processes for setting up HP ALM Platform.
HP Business Process Testing User Guide	Explains how to use Business Process Testing to create business process tests.

HP ALM Best Practices

Guide	Description
HP ALM Database Best Practices Guide	Provides best practices for deploying HP ALM on database servers.
HP ALM Upgrade Best Practices Guide	Provides methodologies for preparing and planning your HP ALM upgrade.
HP ALM Business Models Module Best Practices Guide	Provides best practices for working with the Business Models module.

HP ALM API References

Guide	Description
HP ALM Project Database Reference	Provides a complete online reference for the project database tables and fields.
HP HP ALM Open Test Architecture API Reference	Provides a complete online reference for the HP ALM COM-based API. You can use the HP ALM open test architecture to integrate your own configuration management, defect tracking, and home-grown testing tools with an HP ALM project.
HP ALM Site Administration API Reference	Provides a complete online reference for the Site Administration COM-based API. You can use the Site Administration API to enable your application to organize, manage, and maintain HP ALM users, projects, domains, connections, and site configuration parameters.
HP ALM REST API Reference	Provides an online reference for the HP ALM REST-based API. You can use the REST API to access and work with HP ALM data.
HP ALM Custom Test Type Guide	Provides a complete online guide for creating your own testing tool and integrating it into the HP ALM environment.

Searching and Navigating the Documentation Library

The following functionality is available from the Documentation Library:

Option	Description
	Search and Navigate. Displays the navigation pane. This button is displayed only when the navigation pane is closed. The navigation pane includes the following tabs: ➤ **Contents tab.** Organizes topics in a hierarchical tree, enabling you to directly navigate to a specific guide or topic. ➤ **Search tab.** Enables you to search for specific topics or keywords. Results are returned in ranked order. You can limit your search to a specific guide by selecting a value from the scope drop-down list. **Note:** The search looks for each individual word in the phrase and not for full phrases, regardless of whether you use quotations ("). ➤ **Favorites tab.** Enables you to bookmark specific topics for quick reference. The **Favorites** tab is available only when using the Java implementation of the Help. If your browser does not support Java, the JavaScript implementation is automatically used and the Favorites tab is not displayed.
	Show in Contents. Displays the **Contents** tab in the navigation pane, and highlights the entry corresponding to the currently displayed page. This button is displayed only when the navigation pane is open.
	Previous and Next. Navigates to the previous or next page in the currently displayed guide.
	Send Documentation Feedback to HP. We welcome your feedback. Use this button in any topic to open an email addressed to us, containing the page reference. Send us your comments, ideas for improvement, and any errors you find.

Option	Description
	Print. Prints the currently displayed page. To print a complete guide, access the printer-friendly link from the Documentation Library Home page.
Back	You can use your browser's **Back** function to return to the previously displayed page. In most browsers, you can right-click and select **Back** from the shortcut menu.
Using This Documentation Library	Located on the lower-left corner of each content page. Opens this section.
Glossary	Located on the lower-left corner of each content page. Opens a glossary containing definitions of terms and acronyms.

Topic Types

The content in this guide is organized by topics. Three main topic types are in use: **Concepts**, **Tasks**, and **Reference**. The topic types are differentiated visually using icons.

Topic Type	Description	Usage
Concepts	Background, descriptive, or conceptual information.	Learn general information about what a feature does.
Tasks	**Instructional Tasks.** Step-by-step guidance to help you work with the application and accomplish your goals. Task steps can be with or without numbering: ➤ **Numbered steps.** Tasks that are performed by following each step in consecutive order. ➤ **Non-numbered steps.** A list of self-contained operations that you can perform in any order.	➤ Learn about the overall workflow of a task. ➤ Follow the steps listed in a numbered task to complete a task. ➤ Perform independent operations by completing steps in a non-numbered task.
	Use-case Scenario Tasks. Examples of how to perform a task for a specific situation.	Learn how a task could be performed in a realistic scenario.

Topic Type	Description	Usage
Reference	**General Reference**. Detailed lists and explanations of reference-oriented material.	Look up a specific piece of reference information relevant to a particular context.
	User Interface Reference. Specialized reference topics that describe a particular user interface in detail. Selecting **Help on this page** from the Help menu in the product generally open the user interface topics.	Look up specific information about what to enter or how to use one or more specific user interface elements, such as a window, dialog box, or wizard.
Troubleshooting and Limitations	**Troubleshooting and Limitations**. Specialized reference topics that describe commonly encountered problems and their solutions, and list limitations of a feature or product area.	Increase your awareness of important issues before working with a feature, or if you encounter usability problems in the software.

Additional Online Resources

The following additional online resources are available from the HP ALM **Help** menu:

Part	Description
Troubleshooting & Knowledge Base	Opens the Troubleshooting page on the HP Software Support Web site where you can search the Self-solve knowledge base. Choose **Help** > **Troubleshooting & Knowledge Base**. The URL for this Web site is http://h20230.www2.hp.com/troubleshooting.jsp.
HP Software Support	Opens the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help** > **HP Software Support**. The URL for this Web site is www.hp.com/go/hpsoftwaresupport.
	Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.
	To find more information about access levels, go to:
	http://h20230.www2.hp.com/new_access_levels.jsp
	To register for an HP Passport user ID, go to:
	http://h20229.www2.hp.com/passport-registration.html
HP Software Web site	Opens the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help** > **HP Software Web site**. The URL for this Web site is www.hp.com/go/software.
Add-ins Page	Opens the HP Application Lifecycle Management Add-ins Page, which offers integration and synchronization solutions with HP and third-party tools.

Documentation Updates

HP Software is continually updating its product documentation with new information.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to the HP Software Product Manuals Web site (http://h20230.www2.hp.com/selfsolve/manuals).

1

Introduction to Service Test Management

This chapter includes:

Concepts

➤ Service Test Management Overview on page 20

Concepts

🍀 Service Test Management Overview

When developing an application comprised of many components, it is essential that you organize the components and keep track of their tested status. Service Test Management, in conjunction with other HP testing tools, enables you to test and organize the application components. You can ensure that they perform their intended function and that they are able to scale to the needs of the applications that use them.

Service Test Management's **Application Components** module in ALM, enables you centrally manage your assets. Using this module, you can define and store application components in ALM's repository. After you define components, you can generate a set of requirements and tests to validate the functionality, interoperability, security, boundaries, standards compliance, and performance of the components in your application. You can then run tests and view results in the **Test Plan** and **Test Lab** modules.

Service Test Management also provides change analysis that highlights changes in the components and evaluates the impact that they may have. Based on these changes, Service Test Management can automatically generate the test sets that are required to validate the component.

This section also includes:

➤ "The Service Test Management Integrated Solution" on page 21

➤ "The Advantages of Service Test Management" on page 21

🔵 The Service Test Management Integrated Solution

Service Test Management is an integrated solution that allows you to manage the quality of your application components, such as .NET, Java, or Web Service applications. It integrates the following HP products:

➤ **HP ALM.** Enables you to centrally manage SOA assets from the **Application Components module**. You can define, import, and store services in ALM (Application Lifecycle Management), and generate the necessary requirements and tests.

➤ **HP Service Test.** Enables you to automatically create scripts for testing your SOA environment. You can edit tests generated using Service Test Management, run tests, view results, reuse components and update services.

➤ **HP LoadRunner/HP Performance Center.** Enable you to perform load testing on services, and monitor and predict their performance under load. HP LoadRunner Analysis allows you to drill down to determine the specific source of bottlenecks and generate actionable reports.

➤ **HP Systinet Registry.** Enables you to import Web services from the HP Systinet Registry (supports Systinet 2 with Service Pack 1 or later).

🔵 The Advantages of Service Test Management

Service Test Management provides the following features:

➤ A complete testing process that includes functional testing and performance testing.

➤ Automatically generated requirements and tests that validate the functionality (positive and negative), interoperability, standards compliance, security, and performance of an application component or service.

➤ Reusable tests that can be combined and extended across teams, providing easier transfer of knowledge and test data.

➤ Change management and change impact testing that highlights changes in application components and environments, and evaluates the impact that changes may have on different applications.

21

2

Creating Application Components

This chapter includes:

Concepts

➤ Creating Application Components Overview on page 24

Tasks

➤ How to Define Components on page 29

Reference

➤ ALM Direct Links on page 31

➤ Application Components Module User Interface on page 32

Concepts

Creating Application Components Overview

You create an application components tree with folders, components, and component groups. You can then define the components through contracts.

Once you define application components, you can link them to requirements. This linkage allows you to connect the component to other ALM entities, such as tests, test sets, and defects.

This section also includes:

➤ "The Application Component Tree Hierarchy" on page 27

Application Component Design

Stage 1 - Prepare to collaborate

The QA (Quality Assurance) and Dev (Software Development) managers prepare to collaborate in describing the components or the AUT (Application Under Test). QA will use that information to track changes, execute tests, and report on the quality of the product.

Stage 2 - Set a methodology

Set a methodology to describe the application:

➤ What are the application areas in the AUT, to be translated into folders in the component hierarchy?

➤ What will constitute a component. For example, only high-level interfaces may be considered components, or its centrality or importance in the application?

➤ What component types to define? This could be based on technology, usage, and so forth.

➤ What fields (and possibly UDF) to associate with the component types?

Stage 3 - Quality policies

Determine quality policies to be implemented through rules and testing aspects. These policies are necessary to insure that components are tested based with the correct data.

Stage 4 - Create a project and customize it

Create an Service Test Management project or enable the extension on an existing ALM project. Open Customization and define component types and their fields, as well as UDFs for component and changes. Define the necessary rules and testing aspects.

Stage 5 - Create the application components

Open the **Application Components** module and create a folder structure to represent the application areas decided on above. Create component groups for cross-area functionality or deployment. If possible, define high-level components based on the planned strategy. If not possible, the developers can define them at a later time.

The project is ready for the QA and the software development teams may begin using it.

Stage 6 - Report changes to QA

Developers use the IDE add-on to report changes back to QA. Developers implement requirements and fix defects. They can create new components or application areas as required. These actions will be reported to Service Test Management and assigned to an existing or new application component. They can also be linked to the work item such as a requirement or defect, if one exists.

When making changes, developers describe their risk to QA, to assist them in their testing.

Developers may report a change while still in the process of enacting the change. This allows QA to prepare for testing the change. Optionally, the developers can wait until the change has been completed before reporting it to QA.

Stage 7 - QA processes changes

Before changes are gathered, QA can begin reviewing the application components and assign requirements and tests.

QA engineers can create new component groups to facilitate the management of the components.

After changes are generated, Service Test Management marks the changed components with alert icons and lists them in the Changes tab. QA uses the information in the Interaction tab to create tests for the changed components, using the change description to verify the coverage. During this process, QA can set the following fields:

➤ **Change status** to *In test*.

➤ **Assigned to developer** to determine the developer regarding an aspect of the component's functionality.

➤ **Created by** in the Change tab, to determine the developer who is responsible for updating the component's functionality.

QA can see the changes for one or more components:

➤ Select a component to view the changes specific to that component, and review its **Changes** tab.

➤ Select a group or folder to view an aggregated list of changes in the **Changes** tab. The Changes grid provides the context of the change which can assist QA in understanding the issue.

Stage 8 - Run tests

QA executes tests on the AUT in the testing environment. Based on the test results, they can set the status to **verified** or keep the **In test** status. if a change was tested it is considered **verified**, even if causes defects in the component.

Stage 9 - Reporting

QA uses the graphs and reporting capabilities to summarize the status of the components of the AUT and the changes. They send this report back to the development team. The developers can decide how to act upon QA's recommendations.

The on-going reporting should show a trend of more changes being verified in less time, and a downward trend in the number of defects on the components.

In conclusion, the **Application Components** module allows QA and the developers to verify the quality of the AUT in a comprehensive fashion.

The Application Component Tree Hierarchy

Service Test Management provides you with tools to define components and modify their details.

You organize your application components in a graphical hierarchy.

At the top level of the tree is the **Application Components** root folder, which contains sub folders. You can create custom folders that represent the structure of components in your application. For example, you might create separate folders to represent different types of users.

You can reorganize related entities in folders and subfolders, and drag and drop folders or subfolders to other locations in the tree.

The **Obsolete** folder contains components that were removed, but are still being used by tests stored in ALM. For details about the Obsolete folder, see "Remove and restore components - optional" on page 94.

You cannot rename, move, or delete the **Application Components** root or **Obsolete** folders.
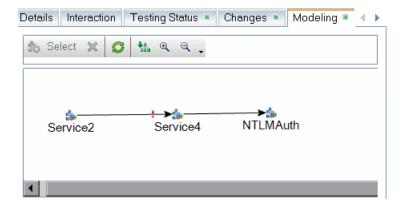
Component Dependencies

Component dependency defines a relationship between two or more components. As the operations of these linked components change, you can detect changes and track their impact on the linked components.

You can define the dependencies using Service Test Management's **Modeling** tab. The **Modeling** tab displays a diagram of the component dependencies.

An arrow indicates a dependency:

➤ An arrowhead pointing to the base component, indicates a **Dependency From** another component—the added component impacts the base component.

➤ An arrowhead pointing away from the base component indicates a **Dependency To** another component—-the added component is impacted by the base component.

You select components and add them to a modeling diagram. In the following example, the **NTLMAuth** component is dependent on **Service4,** while **Service4** is dependent on **Service2**.

Although you add dependent components onto the canvas, the dependencies are per operation. For details, see "Define dependencies - optional" on page 29.

Tasks

🪄 How to Define Components

This task describes how to set up your components in Service Test Managements. After you set them up, you can test them and check their functionality and compliance.

➤ "Create a folder - optional" on page 29

➤ "Create a new component" on page 29

➤ "Specify a contract" on page 29

Create a folder - optional

Click the **New Folder** button to open the New Folder dialog box. Add new folders and subfolders to create a tree hierarchy for your components. For details, see the "New Folder Dialog Box" on page 50.

You can create folders at any time and drag or copy components from folder to folder.

Create a new component

Select **Components** > **New** > **New Component** to create an new component. Select a component type and provide whatever details are available to you. For details, see the "New Application Component Dialog Box" on page 51.

Specify a contract

In the New Application Component Dialog Box, click the **Interaction** link in the left pane. Click **Browse** to locate a contract file. For details, see the "New Application Component Dialog Box" on page 51.

Define dependencies - optional

1 Select the **Modeling** tab. For a description, see the "Modeling Tab" on page 43.

2 Click the **Select Entities** button to open the Components list in the right pane. Expand the folders and select a component.

3 Click the arrow to open the transfer menu.

    a To add a component that is dependent on the base component, click **Add Component Dependency (Dependency To)**.

    b To add a component upon which the base component is dependent, click **Add Component Dependency (Dependency From)**.

The Dependency Details dialog box opens. Select the operation that affects the dependency. If you do not know which operation to select, specify all of them. For details, see the "Dependency Details Dialog Box" on page 49.

Reference

🔍 ALM Direct Links

You can use direct links to access specific views and components. You enter the direct link into your browser to access the desired view. You can save these links for future reference, or send them to others by e-mail so that they may access a specific view or component.

When working with Systinet, you can use a direct link to access the import screen, enabling you to locate services quickly.

The direct links use the following format:

td://<project_name>.<domain>.<server:port>/qcbin/

The following table describes the shortcut syntax to be added to the above string. Italic text indicates optional parameters.

View	Description
Systinet Import	100?Action=ImportSystinet&SystinetID= [SYSTINET_KEY]&Environment=<ENVIRONMENT>&Uui d=<UUID>

View	Description
Component	StmModule--000000000000031130674279? EntityType=IStmService&EntityID=<component id>&*View=<service tab type>* <component tab type> values: ➤ **Changes** ➤ **Modeling** ➤ **TestingStatus** ➤ **AspectStatus** ➤ **OperationStatus** ➤ **StatusDashboard**
Component Group	StmModule--000000000000031130674279? EntityType=IStmGroup&EntityID=<group id>&*View=<group tab type>* <group tab type> values: ➤ **Modeling** ➤ **TestingStatus** ➤ **AspectStatus** ➤ **StatusDashboard**

Modeling Tab on page 43 🔍 **Application Components Module User Interface**

Service Test Management provides a user interface to organize your application components and customize the ALM project hosting them.
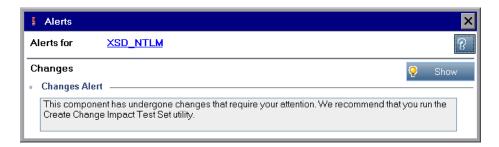
This section includes:

➤ Alerts Dialog Box on page 33

➤ Details Tab on page 35

➤ Interaction Tab on page 38

➤ Find Dialog Box on page 41

➤ Modeling Tab on page 43

➤ Dependency Details Dialog Box on page 49

➤ New Folder Dialog Box on page 50

➤ New Application Component Dialog Box on page 51

➤ Application Components Module Icons on page 55

➤ Application Components Module Window on page 56

➤ Application Components Module Menus and Buttons on page 58

🖉 Alerts Dialog Box

This dialog box displays all alerts and rule violations associated with the selected component and enables you to resolve or disregard them.

To access	Click the red exclamation point ❗ adjacent to the component name in the application components tree.
Relevant tasks	"Evaluate the tests" on page 199
See also	"Change Analysis" on page 193
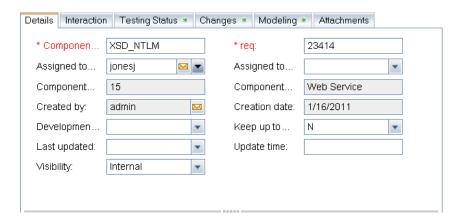
The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
	Removes the current rule violation in the **Rule Violations** pane**.** **Note:** When clearing a rule violation, Service Test Management disregards it—it does not delete it.
	Clears all rule violations in the **Rule Violations** pane: **Note:** When clearing the rule violations, Service Test disregards them—it does not delete them.
	For change alerts, this button opens the **Changes** tab listing all of the changes. Use the **Create Test Set** utility to resolve the changes. For details, see the "Create Change Impact Test Set Wizard" on page 210.
	For rule violations, this button opens the **Create Requirements Test** wizard in order to resolve the violation. For details, see the "Generate Requirements and Tests Wizard" on page 112.
Alerts for <component name>	A link to the component in the application components tree.
Changes	An expandable/collapsible list of all change alerts specific to the selected component. Indicates a change to the component file since its last import or update.
Rule Violations	An expandable/collapsible list of all rule violations for the selected component. A common rule violation is missing coverage for a testing aspect.

🔩 Details Tab

This tab displays and enables you to edit a component's details.

To access	Do the following:
	1 Open the **Application Components** module,
	2 Highlight a folder, component, or group
	3 Click the **Details** tab.
Important information	The Details tab fields differ, depending on the entity you select: folder, component, or group, and the component type.
Relevant tasks	"Manage component details - optional" on page 93
See also	"Application Component Details Pane" on page 46

Details Tab for Components

The following elements are displayed when you select a component in the Component tree.

UI Elements (A-Z)	Description
<user-defined>	User-defined fields defined in customization. The required fields are displayed in red text.
Assigned to Dev	The name of the developer to whom the component implementation is assigned.
Assigned to QA	The name of the QA engineer responsible for validating the component.
Authentication default credentials	Use the default user credentials: **Y** or **N**.
Authentication user name	The user name to use for authentication when accessing the contract file.
Component ID	A read-only ID assigned to the component.
Component name	The native component name as provided by the user.
Component type name	The type of component. The built-in components are **General, Web Service**, and **JMS**. If you defined a new component type, it will be displayed.
Created by	The name with which you logged in. You can edit this field and specify a different name. This is useful for sorting the components in reports.
Creation date	The date the component was created in the Component tree.
Development status	The development status of the component: **Development, Maintenance, QA ready**, or **Stable**.
Keep up to date	Keep the component up to date: **Y** or **N**. When enabled, instructs Service Test Management to update the component from its source each time you select it in the Application Component's tree.

| UI Elements (A-Z)	Description
Last updated	The last time the component was updated.
Visibility	The component's visibility level—**Internal** or **External**.

Details Tab for Groups

The following elements are displayed when you select a group in the application components tree.

UI Elements (A-Z)	Description
Assigned to	The name of the user to whom the group implementation is assigned.
Created by	The name with which you logged in. You can edit this field and specify a different name. This is useful for sorting the groups in reports.
Creation date	The date the group was created in the application components tree.
Description	A description of the group.
Name	The name of the group.
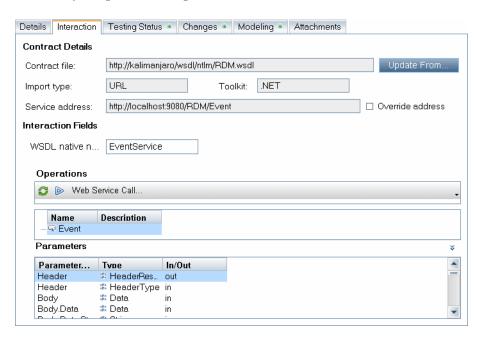
Details Tab for Folders

The following elements are displayed when you select a folder in the application components tree:

UI Elements (A-Z)	Description
Folder ID	A read-only ID assigned to the folder.
Name	The folder name in the application components tree.
Description	A meaningful description of the folder.

✎ Interaction Tab

The **Interaction** tab displays and enables you to edit a component's contract details. The Parameters section shows the parameters and their types, while indicating whether they are used for input or output. For Web services, this tab contains additional information about the WSDL and the **Web Service Call** utility to spot test an operation.

To access	Do the following:
	1 Open the **Application Components** module.
	2 Highlight a folder, component, or group.
	3 Click the **Interaction** tab.
Important information	Only available when selecting a component in the tree. The information differs depending on the component type:
Relevant tasks	"Manage component details - optional" on page 93
See also	"Application Component Details Pane" on page 46

The Interaction tab contains the following information:

UI Elements (A-Z)	Description
Contract Details	➤ **Contract file**. The location of the contract file defining the application component. ➤ Browse.. Opens the Open File or Import Service dialog box—visible for new components whose contracts have not yet been defined. ➤ Update From... . Opens the Open File dialog box. For Web Services, it opens the or Import Service dialog box—visible for components whose contracts have already been defined. For more details, see the relevant section: ➤ "Import Service Dialog Box" on page 78. ➤ "Select Service from Systinet Dialog Box" on page 72 ➤ "Select Service from UDDI Dialog Box" on page 74
Contract Details - Web Services	➤ **Import type**. Contract import method: **File**, **URL**, **Systinet**, or **UDDI**. ➤ **Toolkit**. The toolkit associated with the Web service. You set this in the Customization settings, and it is read-only. ➤ **Service Address**. The service deployment location to which service requests are sent. By default, it displays the endpoint address specified in the WSDL file. ➤ **Override Address**. Enables you to enter an alternate endpoint for the service in the **Service Address** box.
Interaction Fields - Web Service	**WSDL Native Name.** The name of the Service as it appears in the contract file. This section will also include any fields you designated in the Customization for the component type. For details, see "Component Types Tab" on page 181.

UI Elements (A-Z)	Description
Interaction Fields - JMS	➤ **JMS connection factory.** The JNDI name of the JMS connection factory. This setting is unique per test. ➤ **JMS destination.** The name of the JMS destination—either the queue or topic name. ➤ **JMS provider.** The JMS service provider: Apache ActiveMQ, JBoss Messaging from JBoss, Open Message Queue, Websphere MQ from IBM, and so forth. ➤ **JNDI initial context factory.** A The fully qualified class name of the factory class that will create an initial context. Provides a list of context factories and allows manual entries. ➤ **JNDI provider URL.** The URL of the service provider. For example: Websphere - iiop://myserver:myport ➤ **Messaging Model.** The type of JMS message—**Point to Point**, or **Publish and subscribe**.
Operations	➤ 🔄 **Refresh**. Reloads the list of operations. ➤ ▷ Web Service Call... **Web Service Call**. Opens the Run \<operation-name> dialog box to send a test request to the server. For details, see the "Manual Runner Window - Service Testing Tab" on page 83. **Note:** This option is only available if Service Test or the Service Test add-in is installed on the machine. ➤ **\<Operation list>**. A list of the available operations for the application component.
Parameters	A list of the parameters for the selected operation. The list provides the parameter name, data type, and an indication of whether it is an input or output parameter.
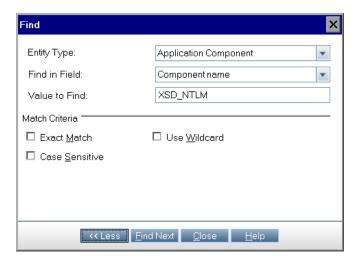
Interaction Tab for UDDI and Systinet Services

For services imported through UDDI or Systinet servers, the **Interaction** tab shows several additional read-only fields:

UI Elements (A-Z)	Description
Systinet Server	The URL address and port of the Systinet server from which the service definition is imported.
UDDI Server	The URL address and version of the UDDI server from which the service definition is imported.
UDDI/Systinet Key	A unique identifier of the service on the UDDI or Systinet server, used to locate the service definition when updating the service.

Find Dialog Box

This dialog box enables you to search for a component, group, or folder within the application components tree.

To access	Select **Edit > Find.**
Relevant tasks	"Find folders and components - optional" on page 93
See also	"Manage component details - optional" on page 93
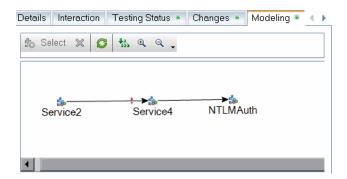
The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements (A-Z)	Description
Find Next	Searches for the next occurrence of the text, based on the previously used settings.
More >>	Expands the Find dialog box to show the match criteria.
<< Less	Collapses the Find dialog box to hide the match criteria.
Entity Type	Type of entity in which to perform the search: **Folder**, **Application Component**, or **Group**.
Find In Field	The field within the Details tab upon which to perform the search. The available fields differ for a folder, group, or component. For information about the fields in the Details tab, see "Details Tab" on page 35.
Match Criteria	Advanced match criteria:, visible when you click **More**. ➤ **Exact Match**. Looks for the exact string—not a partial string. ➤ **Case Sensitive**. Performs a case sensitive search. ➤ **Use Wildcard**. Applies wildcard rules to the search text.
Value to Find	The text to find in the component, group, or folder.

🔧 Modeling Tab

This tab displays component dependencies or component/group relationships in a visual representation. If you select a folder, it shows all associations and dependencies in groups and components beneath that folder.

This tab enables you to associate components with one another and assign them to component groups.

To access	Do the following:
	1 Open the **Application Components** module.
	2 Highlight a folder, component, or group.
	3 Click the **Modeling** tab.
Relevant tasks	➤ "Define dependencies - optional" on page 29
	➤ "Add a component to a group" on page 101
See also	"Component Dependencies" on page 27

User interface elements are described below (unlabeled GUI elements are shown in angle brackets):

UI Elements (A-Z)	Description
⚙ Select	Opens the **Application Components Tree** tab in the right pane to add more component or groups to the modeling window. For information about the tree hierarchy, see the "Application Components Tree" on page 45.
✖	**Delete.** Removes the selected entity from the visual area. You cannot delete the base component or group that is selected in the left pane's **Component** tree.
🔄	**Refresh.** Redraws the visual diagram in the visual area.
🔽	**Arrange layout.** Arranges the layout of all entities in the current modeling window, in a symmetric fashion.
🔍	**Zoom in.** Enlarges the entities in the current drawing.
🔍	**Zoom out.** Reduces the size of the entities in the current drawing.
<modeling canvas>	A visual area showing component dependencies or component/group relationships in a graphical representation.
Application Component/Group Details	Details of the component or group selected in the modeling window. ➤ For component details, see "Application Component Details Pane" on page 46. ➤ For group details, see "Group Details Pane" on page 47.

Application Components Tree

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements (A-Z)	Description
	Add. Opens a drop-down list with the following options: ➤ **Associate.** Associates the selected component with a group. ➤ **Add Component Dependency (Dependency To)** ➤ **Add Component Dependency (Dependency From)** **Note:** For ALM version 11.00 Patch 2 and lower, all options will be enabled. However, the **Associate** option only applies when you select a *group* in the right pane. The **Add Dependency** options is only relevant when you select a *component* in the right pane. For details about dependencies, see the "Dependency Details Dialog Box" on page 49.
	Refresh. Refreshes the application component tree.
	Find. Opens the Find dialog box to search for an application component by its name.
	Set/Clear Filter. Opens a drop-down list with the following options: ➤ **Set Filter / Sort.** Opens the Set Filter/Sort dialog box allowing you to filter and sort the components. ➤ **Clear Filter / Sort.** Clears all filters and sorting settings and displays all components and groups.
	Go To Application Component. Goes to an application component based on its Component ID. The ID is visible in the **Details** tab.

UI Elements (A-Z)	Description
✕	**Exit.** Closes the **Application Components Tree** panel. To reopen this view, click the **Select** Select button in the **Modeling** tab.
<Components List>	An expandable list of all components by folder and group. **Tip:** If a filter is active, there could be hidden components. To see all components, click **Clear Filter**.

## Application Component Details Pane

The following elements are displayed in the bottom pane of the **Modeling** tab, when you select a component in the modeling drawing.

UI Elements (A-Z)	Description
<user-defined>	User-defined fields defined in customization. Required fields are displayed in red.
Assigned to Dev	The name of the developer to whom the component implementation is assigned.
Assigned to QA	The name of the QA engineer responsible for validating the component.
Component ID	A read-only ID assigned to the component.
Component name	The native component name in the definition files.
Component type name	The type of component: **General, Web Service**, and **JMS**. If you defined a new component type in Customization, it will be displayed.
Created by	The name with which you logged in. You can edit this field and specify a different name. This is useful for sorting the components in reports.
Creation date	The date the entity was created in the application components tree.
Development status	The development status of the component: **Development, Maintenance, QA ready**, or **Stable**.

UI Elements (A-Z)	Description
Keep up to date	**Y** or **N**. When enabled, updates the contract from its source each time you select the component in the **Application Component's** tree.
Last updated	The last time the WSDL was updated.
Update time	The time the contract was updated.
Visibility	The visibility of the application component: **Internal** or **External**.

Group Details Pane

The following elements are displayed in the bottom pane of the **Modeling** tab, when you select a group in the modeling drawing.

UI Elements (A-Z)	Description
Assigned to	The name of the user to whom the group implementation is assigned.
Created by	The name with which you logged in. You can edit this field and specify a different name. This is useful for sorting the groups in reports.
Creation date	The date the group was created in the components tree.
Name	The name of the group.
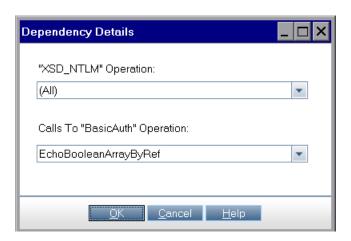
Component Dependency Details Pane

The following elements are displayed in the bottom pane of the **Modeling** tab, when you select a dependency arrow between two components, in the modeling canvas.

UI Elements (A-Z)	Description
New...	**New Dependency.** Opens the Dependency Details dialog box for defining a new dependency between the components. For details, see "Dependency Details Dialog Box" on page 49.
	Edit. Opens the Dependency Details dialog box for editing the entry selected in the Dependency list.
✗ D	**Delete.** Removes the selected dependency.
	Refresh. Reloads the Dependency list.
< **Dependency list**>	A list of the dependencies between the components connected by the selected arrow.

🔍 Dependency Details Dialog Box

This dialog box enables you to set the dependencies between components or modify existing ones.

To access	In the **Modeling** tab:
	➤ For a new dependency: In the Modeling tab, click **Select** to open the list of components in the right pane. Select a component and click the **Add** button ⬅ to create a new dependency. The dialog box opens automatically.
	➤ **For an existing dependency**: Click in the **Component Dependency Details** pane, select a dependency, and click the **Edit** button 📝.
Important information	An arrow indicates a dependency:
	➤ An arrow pointing to the base component indicates a **Dependency From** another component—the added component impacts the base component.
	➤ An arrow pointing away from the base component indicates a **Dependency To** another component—the added component is impacted by the base component.
Relevant tasks	"Define dependencies - optional" on page 29
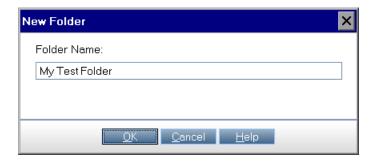
The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements (A-Z)	Description
<component_name> Operation	An operation with which to create a dependency, or **All** if you are unsure.
Calls to <component_name> Operation	The operation through which to create the dependency. Use **Any** if you are unsure.

New Folder Dialog Box

This dialog box page enables you to create a new folder in the application components tree.
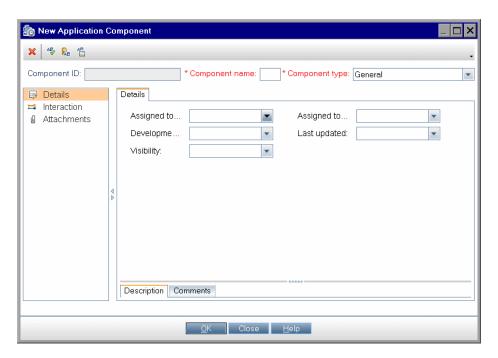
To access	Do one of the following:
	➤ Click the New Folder button.
	➤ Select **Components** > **New** > **New Folder**.
Important information	A folder name cannot include any of the following characters: \ / *
Relevant tasks	"Create a folder - optional" on page 29

User interface elements are described below:

UI Elements (A-Z)	Description
Folder Name	A name for the new folder, beneath the folder selected in the application components tree. You can create several levels of sub-folders. **Tip:** Add a description to the folder in the **Details** tab.
OK	Adds the new folder to the application components tree.

✎ New Application Component Dialog Box

This dialog box enables you to create a new component.

To access	Use one of the following:
	➤ Select **Components** > **New** > **New Component**.
	➤ Select **New** > **New Component** from the shortcut menu.
Relevant tasks	"Create a new component" on page 29

The following elements are included:

UI Elements (A-Z)	Description
✖	**Clear All.** Clears all the fields so that you can redefine the component.
ᴬᴮ✓	**Spell Check.** Begins a spell check in the **Component name** field.
📖ᴬᴮ	**Thesaurus**. Performs a thesaurus lookup for the term highlighted in the **Component name** field.
ᴬᴮ▤	**Spelling Options.** Opens the Spelling Options dialog box for customizing the spell check in the following areas: ➤ **Ignore words in UPPERCASE** ➤ **Ignore words containing numbers** (enabled by default) ➤ **Ignore markup languages (HTML, XML, etc.)** ➤ **Ignore Internet addresses** (enabled by default) ➤ **Prompt on repeated word** (enabled by default) You can also select the main dictionary and a custom one.

UI Elements (A-Z)	Description
![icon]	**Details pane.** The component's properties. The displayed fields differ based on the component type and the fields enabled in the customization section. For details, see the "Component Types Tab" on page 181. Some common fields are: ➤ **Assigned to Dev** ➤ **Assigned to QA** ➤ **Created by** ➤ **Development Status:** Development, Maintenance, QA ready, or Stable. ➤ **Last Updated** ➤ **Visibly:** External or Internal ➤ **Description tab.** An editable area for a description of the component. ➤ **Comments tab.** An editable area for adding remarks.

![icon]	**Interaction pane.** Contract information.
	
	The displayed fields differ based on the component type and the fields enabled in the customization section. To display a field in this section, select it in the customization's **Interaction Field** column. For details, see the "Component Types Tab" on page 181.
	
	➤ **Contract File.** The location of the contract.
	➤ Browse...
	   ➤ For non-Web Service components, opens the Open File dialog box.
	   ➤ For Web Services, opens the Import Service dialog box. For details, see "Import Service Dialog Box" on page 78.
	➤ **Operations.** A list of the component's operations.
	➤ **Parameters.** The component's parameters sorted by the following columns: **In/Out**, **Parameter Name**, and **Type** (data type).

UI Elements (A-Z)	Description
🧰 **for JMS types**	➤ **JMS connection factory.** The JNDI name of the JMS connection factory. This setting is unique per test. ➤ **JMS Destination.** The name of the JMS destination—either the queue or topic name. ➤ **JMS Provider.** The JMS service provider: Apache ActiveMQ, JBoss Messaging from JBoss, Open Message Queue, Websphere MQ from IBM, and so forth. ➤ **JJNDI initial context factory.** The fully qualified class name of the factory class that will create an initial context. Provides a list of context factories and allows manual entries. ➤ **JNDI provider URL.** The URL of the service provider. For example: Websphere - iiop://myserver:myport ➤ **Messaging Model.** The type of JMS message—**Point to point**, or **Publish and subscribe**.
📎	**Attachments pane**. Allows you to attach a file to the component. For details, see the *ALM User Guide*.
Component ID	A read-only value assigned to each application component.
Component name	A name for the component. Use the **Spell Check** or **Thesaurus** buttons for the textual strings.
Component type	The type of component to create: **General**, **Web Service**, or **JMS**. If you created a custom type component in Customization, it also appears in the drop-down list.

✎ Application Components Module Icons

This section describes the icons available in the **Application Components** module.
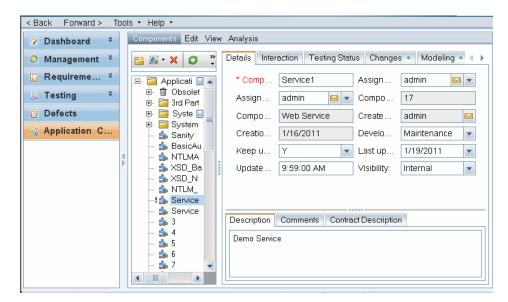
To access	Click the **Application Components** button on the ALM sidebar.
See also	➤ "Application Components Module Window" on page 56
	➤ "Application Components Module Menus and Buttons" on page 58

The icons are described below:

UI Elements	Description
!	**Alert.** Click to view a summary of the following alert types:
	➤ **Changes.** The component definition has changed.
	➤ **Rule Violations.** The component violates a rule.
	For more details, see the "Alerts Dialog Box" on page 33.
	Folder. A folder for organizing components.
	Obsolete folder. Contains components that were deleted that may affect other components.
	General type. A General type application component.
	Web Service type. A Web Service type application component.
	JMS type. A JMS type application component.
	Application Component group.

🔍 Application Components Module Window

This page describes the **Application Components** module window.

To access	Click the **Application Components** button on the ALM sidebar.
Relevant tasks	"Manage component details - optional" on page 93
See also	➤ "Application Components Module Icons" on page 55 ➤ "Application Components Module Menus and Buttons" on page 58

User interface elements are described below:

UI Elements (A-Z)	Description
<toolbar and menu>	Provides menus and toolbar buttons specific for the **Application Components** module. For details, see the "Application Components Module Menus and Buttons" on page 58.
<Components Tree>	Shows a tree hierarchy of all the components, component groups, and folders. For details, see "Application Components Module Icons" on page 55.
Attachments tab	Lists the files that have been associated with the selected folder, component, or group as attachments. The tab includes an icon ✳ if the selected component has attachments. For details, see the *HP ALM User Guide*.
Details tab	Displays general details and attributes of the selected component, group, or folder. Displays data that is required to identify the component, including the creation information. For details, see the "Details Tab" on page 35.
Changes tab	Lists the changes in the components. It also provides an interface to define new changes that you made to the components and create test sets to check their impact. For details, see "Reporting Changes by Updating Components" on page 195.
Interaction tab	Provides the component's contract details, such as the WSDL location for Web Services. For details, see the "Interaction Tab" on page 38.
Modeling tab	Provides an interface to graphically associate components with groups, and to define dependencies between them. For details, see "Modeling Tab" on page 43.
Testing Status tab	Provides a coverage summary for aspects, requirements, operations, tests, rule violations, and defects. For details, see "Determining Test Coverage" on page 119.

🔧 Application Components Module Menus and Buttons

This section describes the menus and buttons available in the **Application Components** module.

To access	Open the **Application Components** module on the ALM sidebar.
Important information	All menu items are available from the right-click shortcut menu.
Relevant tasks	"Manage component details - optional" on page 93

User interface elements are described below:

UI Elements (A-Z)	Menu	Description
📂	Toolbar	**New Folder.** Opens the New Folder dialog box to create a folder under the root or an existing folder.
📥	Toolbar	**New Component.** Opens the New Application Component dialog box. **Note:** This area on the toolbar displays the last button used: **New Component** or **New Group**.
📥	Toolbar	**New Group**. Opens the New Group dialog box.
✖	Toolbar	**Delete**. Deletes selected components. Deleting a component also deletes its children. You cannot delete the root folder.
🔄	Toolbar	**Refresh All.** Refreshes the application components tree so that it displays the most up-to-date versions.

UI Elements (A-Z)	Menu	Description
▼ ▾	Toolbar	**Filter/Sort.** Enables you to filter and sort the requirements in the requirements tree or grid. For more details, see below.
✉	Toolbar	**Send By E-mail.** Opens the Send E-mail dialog box, enabling you to send component URLs and other information to recipients from a list or to the creator of the component. Use this in conjunction with **Copy URL.** For more details, see the *HP ALM User Guide*.
Copy URL/Paste	Edit	Copies a selected component and pastes its URL as a link. The component itself is not copied. Instead, you can paste the address into another location, such as an e-mail or a document. Clicking the link opens ALM and takes you to the requirement. If you are not already logged in, ALM first prompts for login details.
Copy/Paste	Edit	Copies selected components within the same project or across projects. When you copy a folder, its sub-folders are copied too. **Notes:** ➤ Coverage and modeling data for the component are not copied. ➤ If you paste a component that has the same name as an existing component, the suffix _Copy_x is added automatically to the end of the name.

UI Elements (A-Z)	Menu	Description
Create Empty Test	Components	Opens the Create Empty Test dialog box, prompting you for a location and name. This is useful for testers who know the required steps, but are unfamiliar with the technical aspects of creating tests.
Cut/Paste	Edit	Moves selected component to a different location in the application components tree. **Tip:** You can also move a component to a new location by dragging it.
✖ **Delete**	Edit	Deletes the selected component.
Download Contract File	Components	Allows you to download a contract file from ALM to a local file system.
Expand All/Collapse	View	Expands all or collapses the selected branches in the application components tree.
▼ **Filter/Sort**	View	Enables you to filter and sort application components in the tree. For more details, click the **Help** button.
Find	Edit	Searches for a component in the **Application Components** module. For details, see "Find Dialog Box" on page 41.
Generate Requirements/ Test	Components	Opens the Generate Requirements and Test wizard. For details, see "Generating Requirements and Tests" on page 105.

UI Elements (A-Z)	Menu	Description
Graphs	Analysis	Generates one of the following graphs: ➤ **Components Summary - Group by 'Type'** ➤ **Changes Progress - Group by 'Risk'** ➤ **Changes Summary - Group by 'Risk'** ➤ **Changes Trend - Group by 'Status'**
New	Components	Enables you to create a new entity: ➤ **New Folder.** Enables you to create a new folder in the application components tree. ➤ **New Group.** Opens the New Group dialog box. ➤ **New Component.** Opens the New Component dialog box.
Refresh All	View	Refreshes the Component tree so that it displays the most up-to-date versions.
Rename	Edit	Renames the selected entity. You cannot rename the root folder. **Syntax exceptions:** ➤ Folder or group names cannot include: *^. ➤ Component names cannot include: \, /, :, ", ?, ', <, >, |, *, %, !, {, or }.
Restore Component	Components	Restores a deleted component from the Obsolete folder.

UI Elements (A-Z)	Menu	Description
✉ **Send by E-mail**	Components	Opens the Send E-mail dialog box, enabling you to send URLs and other information to recipients from a list or to the creator of the component. Use this in conjunction with **Copy URL.** For more details, see the *HP ALM User Guide*.
Update Component	Components	Enables you to update a component from: ➤ **Update Component.** Updates the component from its original location. ➤ **Update Component from.** Opens a dialog box for navigating to a URL or a file with an updated version of the contract.

3

Web Services

This chapter includes:

Concepts

➤ Defining Web Services Overview on page 64

➤ Creating Tests in Service Test on page 67

Tasks

➤ How to Define a Web Service on page 68

Reference

➤ Web Service Specific User Interfaces on page 71

Concepts

🔩 Defining Web Services Overview

Web services are a specific case of application components, where the contract is a WSDL file, defining the Web service, its address, and the operations. Service Test Management's functionality for application components contains several additional features specific for Web Services, such as Spot Testing and specialized reports.

When working with Web Services, you create a component hierarchy as described in "Creating Application Components Overview" on page 24.

Web Service contracts are typically WSDL files. Service Test Management allows you to import WSDL files that define the Web service and its operations.

You can import a definition from a file location, URL, UDDI or Systinet server. When you import a definition, Service Test Management extracts the service name, service address (endpoint), and data from the definition file, and stores it in the ALM repository.

For defining application components that are not WSDL-based, see the "New Application Component Dialog Box" on page 51.

This section also includes:

➤ "Importing Services" on page 65

➤ "Secure Services and Proxy Servers" on page 65

➤ "Spot Testing Operations" on page 66

🔷 Importing Services

You can import Web service definitions based on WSDL files from a file location, URL, UDDI or Systinet server. When importing a WSDL file, Service Test Management parses the XML code and stores the service in the ALM repository.

You can then link the service to other ALM entities such as tests, test sets, and defects. These links help you to keep track of the service and ensure compliance with your requirements throughout the testing process. For details about requirement coverage, see "Generating Requirements and Tests" on page 105.

You can specify connection settings for WSDL files that reside in secure locations or accessible through proxy servers. These settings let you specify authentication and proxy server details. Once you enter the security or proxy information, it remains permanently associated with the service, enabling automatic updates. For details, see "Secure Services and Proxy Servers" on page 65.

For user interface details, see the "Import Service Dialog Box" on page 78.

Toolkit Support

When setting up the import, Service Test Management lets you specify a toolkit to match the development environment. Service Test Management supports the .NET and Axis toolkits. The default **Auto-detect** option attempts to parse the selected WSDLs using the available toolkits, according to the parsing order set in Project Customization.

For details about customizing the toolkit parsing order, see the "Customization Overview" on page 162.

🔷 Secure Services and Proxy Servers

When importing WSDL files from a URL or UDDI server, the WSDL may require authentication if it resides in a secure location. Certain WSDLs may only be accessed through a proxy server.

Service Test Management supports WSDLs that require authentication and WSDLs accessed through proxy servers, using basic and NTLM authentication.
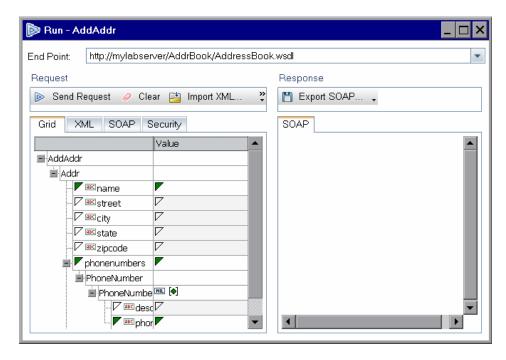
The security or proxy information is always associated with the WSDL, If you enable the **Keep up to date** option to allow automatic synchronization, Service Test Management uses the authentication or proxy server settings to access the WSDL at its source.

Note: Setting default connection settings in the project customization area, instructs Service Test Management to use the same settings for all your services. For details, see "How to Customize a Project" on page 167.

🔧 Spot Testing Operations

The **Web Service Call** utility lets you test your operations one by one, also known as *spot testing*. This utility lets you check that the operation is functional and that its response is valid.

You use the grid to enter request values and submit them to the server. You can view the SOAP response and export it to a file.

For task details, see "Test the service - optional" on page 70.

For user interface details, see the "Run <Operation> Dialog Box" on page 80.

Creating Tests in Service Test

If you have HP Service Test, you can create tests to test the components in Service Test Management.

In Service Test, select **File > ALM/QC Connection** to connect to the ALM project that contains your Web Service type components.

Create a test in the normal way, and save it to the appropriate location in the ALM repository.

Use the **Application Components** tab in the **Test Plan** module to link a component to the test. For details, see "Application Components Tab - Test Plan Module" on page 156.

For more information, see the *HP Service Test User Guide*.

Tasks

🪝 How to Define a Web Service

This task describes how to set up your components in Service Test Management. After you set them up, you can test them and check their functionality and compliance.

➤ "Create a folder" on page 68

➤ "Create a new component" on page 69

➤ "Import a service" on page 69

➤ "Set authentication or proxy information - optional" on page 69

➤ "Create dependencies - optional" on page 69

➤ "Test the service - optional" on page 70

➤ "Run the test" on page 70

1 Create a folder

The development team and Quality Assurance should design a tree hierarchy representing the structure of the application. Open the New Folder dialog box and add new folders and subfolders based on your design requirements. For details, see the "New Folder Dialog Box" on page 50.

You can create folders at any time and drag or copy components from folder to folder.

2 Create a new component

Create a new component. Select **Components** > **New** > **New Component**. For details, see the "New Application Component Dialog Box" on page 51.

3 Import a service

In the New Application Component Dialog box, open the **Interaction** view and click **Browse** to open the Import Service dialog box. Import a Web service. For details, see the "Import Service Dialog Box" on page 78.

4 Set authentication or proxy information - optional

If the WSDL file requires authentication or if it accessible through a proxy server, click the **Connection Settings** button to configure the connection settings. For details, see the "Connection Settings Dialog Box" on page 76.

5 Create dependencies - optional

Dependencies indicate the reliance of one component upon another.

a Select the **Modeling** tab. For a description, see the "Modeling Tab" on page 43.

b Click the **Select Entities** button to open the Components list in the right pane. Expand the folders and select a component.

c Click the arrow to open the transfer menu.

> ➤ To add a component that is dependent on the base component, click **Add Component Dependency (Dependency To)**.

> ➤ To add a component upon which the base component is dependent, click **Add Component Dependency (Dependency From)**.

The Dependency Details dialog box opens.

d Select the operation that affects the dependency. If you do not know which operation to select, specify all of them. For details, see the "Dependency Details Dialog Box" on page 49.

6 **Test the service - optional**

To generate a Web Service call in order to test your Web Service, click the **Interaction** tab, select an operation, and click **Web Service Call**. In the grid, enter request values. Send the request to the server and check the SOAP response. For details, see the "Run <Operation> Dialog Box" on page 80.

7 **Run the test**

After you verify that your service is valid, run the test or test set as you would run any test in ALM, from the **Test Lab** module. For details, see the *ALM User Guide*.

The manual running option provides an interface for working with Web Services. For details, see the "Manual Runner Window - Service Testing Tab" on page 83.

Reference

🔍 Web Service Specific User Interfaces
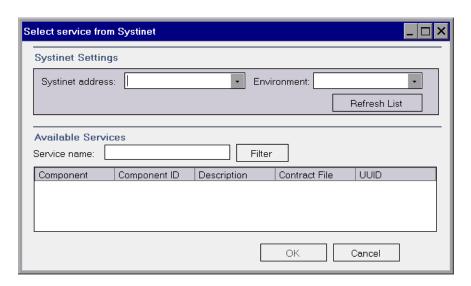
Service Test Management provides a user interface to import Web Service contracts and test the Web Service Calls.

This section includes (in alphabetical order):

➤ Select Service from Systinet Dialog Box on page 72

➤ Select Service from UDDI Dialog Box on page 74

➤ Connection Settings Dialog Box on page 76

➤ Import Service Dialog Box on page 78

➤ Run <Operation> Dialog Box on page 80

➤ Manual Runner Window - Service Testing Tab on page 83

🔍 **Select Service from Systinet Dialog Box**

This dialog box enables you to select and import services from a Systinet server.
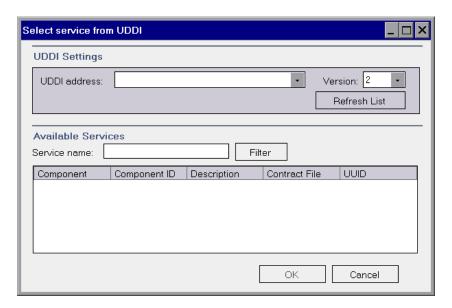
To access	In the Import Service dialog box, choose **Import WSDL from**: **Systinet** and click the **Browse** ⋯ button.
Important information	**Note:** Only ALM Governance 4.0 (Systinet 4.0) is supported for Service Test Management 11.00.
Relevant tasks	"Import a service" on page 69
See also	➤ "Import Service Dialog Box" on page 78 ➤ "Select Service from UDDI Dialog Box" on page 74

The following elements are included:

UI Elements	Description
Refresh List	Reloads the list of the services on the Systinet server, in the **Available Services** pane.
Filter	Filters the list of services to show only those services containing the string specified in the **Service name** field.
Systinet Address	The name or IP address of the Systinet server. **For Example:** http://pumpkin:8080/soa
Environment	The Systinet environment and version.
Service name	A string in the service name by which to filter the list.
Available Services	A list of the services that match the filter criteria in alphabetical order. The grid shows the following information: ➤ **Component Name** ➤ **Component ID** ➤ **Description** ➤ **Contract File** ➤ **UUID** (Universally Unique Identifier)

🔍 Select Service from UDDI Dialog Box

This dialog box enables you to select and import services from a a UDDI (Universal Description, Discovery, and Integration) server.

To access	In the Import Service dialog box, choose **Import WSDL from**: **UDDI** and click the **Browse** ⊡ button.
Relevant tasks	"Import a service" on page 69
See also	"Select Service from Systinet Dialog Box" on page 72
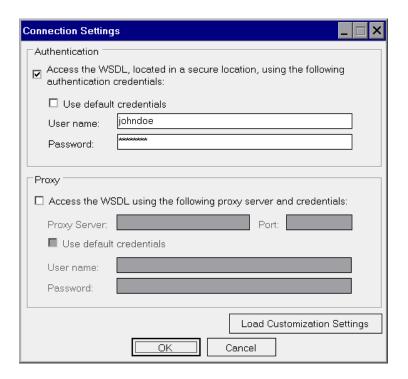
The following elements are included:

UI Elements	Description
Refresh List	Reloads the list of the services from the UDDI, in the **Available Services** pane.
Filter	Filters the list of services to show only those services containing the string specified in the **Service name** field.

UI Elements	Description
UDDI address	The name or IP address of the UDDI server inquiry API. **For Example:** http://lab1.devlab.ad:8090/juddi/inquiry.
Environment	The UDDI version—2 or 3.
Service name	A string in the service name by which to filter the list.
Available Services	A list of the services that match the filter criteria. The grid shows the following information: ➤ **Component Name** ➤ **Component ID** ➤ **Description** ➤ **Contract File** ➤ **UUID** (Universally Unique Identifier)

🔍 Connection Settings Dialog Box

This dialog box enables you to provide authentication credentials and proxy server details for the machine hosting a Web Service's WSDL file.

To access	➤ For a new service: In the New Component window (**Components** > **New** > **New Component**), select the **Interaction** node and click the **Browse button**. In the Import Service dialog box, lick the **Connection Settings** button.
	➤ For existing services: Select the **Interactions** tab, click **Update from**, and click **Connection Settings**.
	➤ In Customization: Select **Service Test Management** link > **General** Tab > **Default Connection Settings.**
Important information	Available only for services imported through a URL and UDDI.
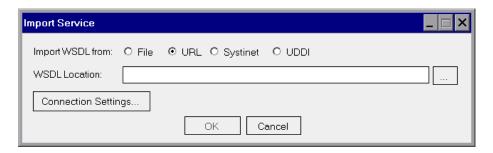
Relevant tasks	"Import a service" on page 69
See also	You can set the default WSDL connection settings in the Project Customization window. For details on customizing the connection settings, see "Customizing Service Test Management" on page 161.

The following elements are included:

UI Elements (A-Z)	Description
Authentication	**Access the WSDL located in a secure location, using the following authentication credentials.** Enables you to enter credentials for authentication: ➤ **Use default credentials.** Authenticates with the Windows logon credentials of the user logged on to the machine. ➤ **User name, Password.** The user name and password to be used for authentication. For users not in the default domain, type the domain name before the user name. For example, domain1/qc_user1.
Proxy	**Access the WSDL using the following proxy server and credentials.** Enables you to enter proxy details and credentials: ➤ **Proxy server.** Name or IP address of proxy server. ➤ **Port.** Port through which to access the WSDL. ➤ **Use default credentials.** Connects to the proxy server with the Windows logon credentials of the user logged on to the machine. ➤ **User name, Password.** The user name and password to be used for authentication. For users not in the default domain, type the domain name before the user name. For example, domain1/qc_user1.
Restore Defaults	Restores the default connection settings—user name authentication and no proxy server.

🔍 Import Service Dialog Box

For Web Service type application components, this dialog box enables you to import WSDLs from a file system, a URL, a UDDI, or Systinet.
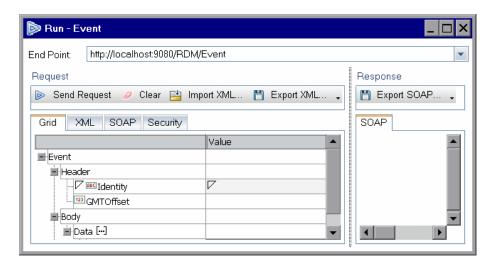
To access	➤ For new Web services: In the New Application Component window (**Components** > **New** > **New Component**), select the **Interaction** link. Click **Browse**. ➤ For existing Web services: Select the **Interaction** tab and click **Update From**.
Relevant tasks	"Import a service" on page 69.
See also	➤ "Connection Settings Dialog Box" on page 76 ➤ "Select Service from Systinet Dialog Box" on page 72 ➤ "Select Service from UDDI Dialog Box" on page 74

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements (A-Z)	Description
`[...]`	Lets you locate the WSDL by opening one of the following: ➤ **File.** An Open File dialog box ➤ **URL.** A new browser window ➤ **Systinet.** Select Service from Systinet dialog box ➤ **UDDI.** Select Service from UDDI dialog box
Connection Settings...	Opens the Connections Settings dialog box for configuring the authentication and proxy settings of the server hosting the WSDL. For details, see "Connection Settings Dialog Box" on page 76.
Import WSDL from	The WSDL source: ➤ **File.** The file system ➤ **URL.** A URL ➤ **Systinet.** A Systinet registry ➤ **UDDI.** A UDDI server
WSDL Location	Location of WSDL: ➤ **File:** full path and file name ➤ **URL:** Complete URL ➤ **Systinet:** Systinet ID ➤ **UDDI**: Server key Use the **Browse** button to locate the WSDL.

🔍 Run <Operation> Dialog Box

This dialog box enables you to run a spot test for an individual operation. You can set up a request and view the SOAP response to verify if an individual operation is functional.

To access	Select an operation in the **Interaction** tab and click the **Web Service Call** button.
Important information	You must have the Service Test add-in or Service Test installed to use this utility.
Relevant tasks	"Test the service - optional" on page 70.

User interface elements are described below.

UI Elements (A-Z)	Description
End Point	The address of the server over which to submit the request.

UI Elements (A-Z)	Description
Request Pane	The request values and controls. For details, see the "Run Operation - Request Pane" on page 81.
Response Pane	The SOAP response. For details, see the "Run Operation - Response Pane" on page 82.

Run Operation - Request Pane

User interface elements are described below:

UI Elements (A-Z)	Description
▷ Send Request	Sends the request to the server address specified in the **End Point** box.
⌀ Clear	Clears the values you assigned to the arguments in the **Grid** tab.
Import XML...	Opens the Import XML dialog box, allowing you to import an XML structured file, for the purpose of loading values for the request.
Export XML...	Opens the Export XML dialog box, allowing you to export the values in the grid to an XML file, for the purpose of reusing them at a later stage.
Grid Tab	A list of all arguments, and a column for setting their values. Click on a triangle adjacent to an argument to include/exclude it from the request. ➤ A filled-in triangle includes the argument in the request; An empty triangle excludes the optional argument from the request. ➤ An argument with an ABC icon can be parameterized. ➤ A NIL icon indicates that argument can have a NIL value.

UI Elements (A-Z)	Description
Security Tab	The security policy: ➤ **No Security Options**. Do not use any security (default). ➤ **Use Authentication Settings**. Authenticate using a user name, password, and password send option—**PlainText**, **None**, or **Hashed**. ➤ **Use Kerberos**. Use the Kerberos token security.
SOAP Tab	The request submitted to the server in SOAP format.
XML Tab	The request submitted to the server in XML format.

Run Operation - Response Pane
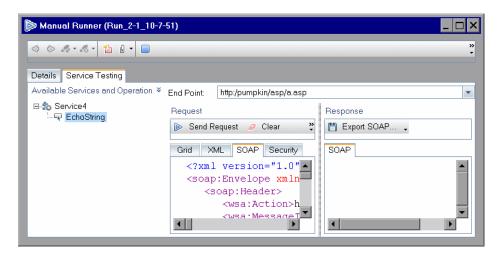
User interface elements are described below:

UI Elements (A-Z)	Description
Export SOAP...	Opens the Export SOAP dialog box, allowing you to export the SOAP response to an XML file.
SOAP Tab	The SOAP response resulting from the request.

🔍 Manual Runner Window - Service Testing Tab

This tab lets you manually run a Web Service test step. This only applies to Web Service components with WSDL contracts.

You can verify the functionality of a specific operation using Spot testing. For details, see "Spot Testing Operations" on page 66.

To access	Perform the following steps:
	➤ Select a test set in the **Test Lab** module's **Test Sets** tab.
	➤ Click the **Execution Grid** tab.
	➤ Select a test and choose **Run** > **Run with Application Component Manual Runner.**
	➤ Click the **Service Testing** tab (by default this is open).
	For application components without WSDL contracts, select **Run with Manual Runner**. For details, see the *ALM User Guide*.
Important information	You must have the Service Test add-in or an installation of HP Service Test to view this tab.
Relevant tasks	"Test the service - optional" on page 70

User interface elements are described below.

UI Elements (A-Z)	Description
◁	**Previous Step.** Navigates to the previous test step.
▷	**Next Step.** Navigates to the next test step.
✓ ▾	**Set Pass Status.** Sets the test status to **Pass**. The drop down list provides the following options: ➤ **Pass Selected.** Sets status to **Pass** for selected steps. ➤ **Pass All.** Sets status to **Pass** for all steps of the test.
✗ ▾	**Set Fail Status.** Sets the test status to **Fail**. The drop down list provides the following options: ➤ **Fail Selected.** Sets status to **Fail** for selected steps. ➤ **Fail All.** Sets status to **Fail** for all steps of the test.
🗒	Opens the New Defect dialog box for creating a defect for the current test and test set.
📎 ▾	**Add Attachment to Run**. Adds an attachment to the test run. An attachment can be a file, URL, snapshot of your application, an item from the clipboard, or system information. The drop down list provides the following options: ➤ **Attachment to Step.** Adds an attachment to the current step. ➤ **Attachment to Run.** Adds an attachment to all steps.
▣	**End Run.** Stops the test run. **Note:** If you stop a test run before it is complete, you can resume the test run from the point at which it was stopped. In the **Execution Grid** tab, select **Run > Continue**.

UI Elements (A-Z)	Description
Actual Result	The actual value returned. If this value matches the Expected Value, mark the status to **Pass**. Otherwise, set the status to the appropriate value. For a Web service step, use the **Send Request** button to check the response.
Available Services and Operations	A list of all services and operations available for the test.
Description	An editable description of the test step.
End Point	The address of the server over which to submit the request. Specify a path to override the WSDL's default.
Expected Result	The value you expect the server to return.
Request Pane	The request values and controls. For details, see the "Manual Runner - Service Testing Tab Request Pane" on page 86.
Response Pane	The SOAP response. For details, see the "Manual Runner - Service Testing Tab Response Pane" on page 87.
Status	The status you want to manually assign to the test: **Fail, N/A, No Run, Not Completed,** or **Pass**.
Step	The step number whose details are displayed.

Manual Runner - Service Testing Tab Request Pane

User interface elements are described below:

UI Elements (A-Z)	Description
▷ Send Request	Sends the request to the server address specified in the End Point box.
	Before sending the request, select the desired operation in the **Available Services and Operations** pane. In the **Grid** tab, provide argument values. If required, modify the XML and then send the request.
⊘ Clear	Clears the values you assigned to the arguments in the **Grid** tab.
🗁 Import XML...	Opens the Import XML dialog box, allowing you to import an XML structured file, for the purpose of loading values for the request.
🖫 Export XML...	Opens the Export XML dialog box, allowing you to export the values in the grid to an XML file, for the purpose of reusing them at a later stage.
Grid Tab	Contains a list of all arguments, and a column for setting their values. Click on a triangle adjacent to an argument to include/exclude it from the request.
	➤ A filled-in triangle includes the argument in the request; An empty triangle excludes the optional argument from the request.
	➤ An argument with an ABC icon can be parameterized.
	➤ A NIL icon indicates that argument can have a NIL value.
Security Tab	Connection and Authentication information required for performing the Web Service call.
SOAP Tab	The request submitted to the server in SOAP format.
XML Tab	The request submitted to the server in XML format.

Manual Runner - Service Testing Tab Response Pane

User interface elements are described below:

UI Elements (A-Z)	Description
💾 Export SOAP...	Opens the Export XML dialog box, allowing you to export the values in the grid to an XML file, for the purpose of reusing them at a later stage.
SOAP Tab	The SOAP response resulting from the request.

4

Managing Application Components

This chapter includes:

Concepts

➤ Managing Application Components Overview on page 90

Tasks

➤ How to Manage Components on page 93

Concepts

🎯 Managing Application Components Overview

After you define an application component, you can view and modify its definitions and details. You can also resolve alerts and view the changes on your component. Using the application components tree, you can reorganize the components under different folders.

This section includes:

➤ "Application Component Details" on page 90

➤ "Locating Application Components" on page 91

➤ "Handling Rule Violations or Change Alerts" on page 91

➤ "Copying Components" on page 91

➤ "Adding Attachments" on page 92

➤ "Restoring and Removing Components" on page 92

➤ "E-mailing Components" on page 92

Application Component Details

In the application components tree, you can view and modify the location of components and folders. For Web service components, you can view information such as the WSDL location, override the service address, and modify connection settings.

For details, see "Manage component details - optional" on page 93.

Locating Application Components

You can search for a particular entity in the application components tree—folder, component, or group, by one of their **Details** tab fields. The search utility allows wildcards and searching by a specific field. For example, you can search for a component that begins with the letter "s".

Tip: If you applied a filter to the tree, the search is restricted to the folders and components currently displayed.

For details, see "Find folders and components - optional" on page 93.

Handling Rule Violations or Change Alerts

Service Test Management uses alert indicators to make you aware of changes to the components or rule violations.

The changes can be those entered manually or those detected automatically by Service Test Management when you update a contract.

Rule violations are a result of non-compliance with the component's requirements. To complete your testing, you should try to resolve the alerts.

For additional information, see "Handle alerts - optional" on page 94.

Copying Components

You can copy an existing folder, component, or group and paste it to another location in the application components tree, or to another project. When copying a component to another project, the source and target projects must be open in separate browsers.

For more information, see "Copy entities - optional" on page 94.

Adding Attachments

The **Attachments** tab enables you to associate an attachment with a folder, component, or group. An attachment can be a file, URL, text from the clipboard, snapshot, or system information. It is identified by its name, associated application icon, size, and latest modification date and time. You can add, modify, and delete attachments.

The **Attachments** tab includes an icon ✳ if the selected folder, component, or group has any attachments.

The **Attachments** tab for folders, components, and groups has the same functionality as the **Attachments** tab for other types. For details on adding attachments, refer to the *HP ALM User Guide*.

Restoring and Removing Components

You can permanently delete folders or components that are not used by a test or requirement. To see if a component is used by a test, see "Tests Coverage" on page 124. If you try to delete components that are used by one or more tests, ALM transfers them to the **Obsolete** folder. You cannot copy obsolete components.

You can restore components in the **Obsolete** folder to their original position in the application components tree if the folder still exists in the tree, or you can drag them to a different location in the tree.

For details, see "Restoring and Removing Components" on page 92.

E-mailing Components

You can send e-mail about an entity (folder, component, or group) to other users. This enables you to routinely inform users about the status of your components. A link is included in the e-mail message that enables the recipient to go directly to the component. You can also include your own shortcuts using one of the formats described in "ALM Direct Links" on page 31.

For additional information, see "Mail components- optional" on page 95.

Tasks

How to Manage Components

This task describes how to work with your components by changing its details, updating a WSDL, resolving alerts, copying entities, and so forth.

This task includes the following steps:

Manage component details - optional

Open the **Details** tab and modify the component's details as desired.

For more information, see "Details" on page 35.

Find folders and components - optional

1 To search all of the application components, select **Filter** > **Clear Filter** to remove any filters.

2 Select **Edit** > **Find**. The Find dialog box opens.

3 Select an **Entity Type**: Folder, Application Component, or Group.

4 Select the field to search in the **Find in Field** box. For example to search for a component by its name, select **Component Name** field and specify a part or the whole name as a value.

5 Specify a value in the **Value to Find** box. The search engine is not case-sensitive and it finds parts of a string.

6 Click **Find Next** to locate the first and subsequent matches.

For details, see the "Find Dialog Box" on page 41.

Handle alerts - optional

Click on an exclamation point adjacent to a component, to view its alerts. Determine which alerts to resolve. For details, see the "Alerts Dialog Box" on page 33.

Copy entities - optional

Using the right-click shortcut menu, copy an existing folders, components, or groups onto the clipboard. Open the target folder in the components tree and paste the entity from the clipboard. To copy it to another project, open the target project in another browser tab and paste it in.

For details, see the "Application Components Module Window" on page 56.

Remove and restore components - optional

To remove a component, select it and click the **Delete** button. If you remove a folder, its subfolders and all components in those folders are removed too. If a removed component is still being used by an existing test, Service Test Management stores it in the **Obsolete** folder.

Tip: To view all application components in the **Obsolete** folder, select the folder and click the **Refresh** button.

To remove a component entirely, make sure that no tests are using it. Click the **Test Script** tab in the **Test Lab** module to open the test in Service Test, and remove all steps calling the component. After you have removed all calls to the component, delete it from the **Obsolete** directory.

You can restore components in the **Obsolete** folder to their original position in the components tree, if the folder still exists in the tree. If the folder no longer exists, transfer the component to another location in the tree by dragging it from the **Obsolete** folder to the desired location.

Mail components- optional

Select a component and click the **Send by e-mail** button. A link included in the e-mail message enables the recipient to go directly to the component.

Edit the content, select recipients, and perform a spell check. For details about the Send Mail dialog box, click F1 or see the *HP ALM User Guide*.

5

Groups

This chapter includes:

Concepts

➤ Groups Overview on page 98

Tasks

➤ How to Manage Component Groups on page 101

Reference

➤ New Groups User Interface on page 103

Concepts

🔩 Groups Overview

In an organization, application components have a context. For example, the components may be located on the same server, may be exposed as part of the same application, or may take part in the same business process.

In the Application Components module, you use groups to organize your components into logical groupings. You might, for example, organize them into groups representing different physical deployments, business processes, or application types such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM). After you create the groups, you can associate components to them, and perform tasks upon all members of the group.

You can add attachments to a group and send e-mails in a similar manner to that of individual components. For details, see the *HP ALM User Guide*.
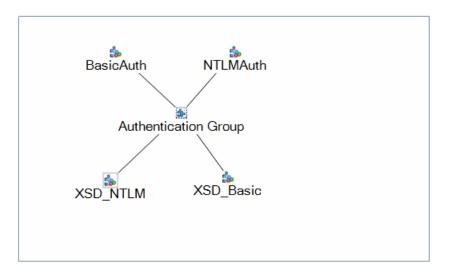
This section also includes:

➤ "Associating Components to Groups" on page 98

➤ "Group Coverage" on page 100

Associating Components to Groups
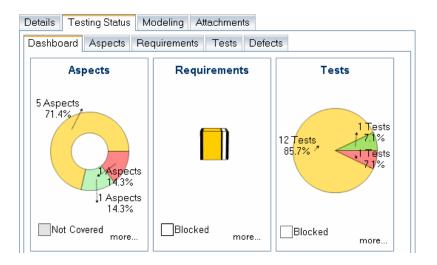
Groups help you organize your components into logical units. You might, for example, organize them into groups representing different physical deployments, business processes, or application types. After you create groups, you can associate components to the group, and perform tasks upon the entire group. The same component can be assigned to multiple groups.

The **Modeling** tab lets you create and view the associations between the groups and components. For details, see the "Modeling Tab" on page 43.

Group Coverage

You define requirements and tests for individual components—not for groups. When you associate components with groups, the group reflects the coverage of all its associated entities. You can view the complete coverage in the group's **Testing Status** tab. Click any of the sections in the **Testing Status** tab, to see the specific coverage for **Aspects**, **Requirements**, or **Tests**.

For details, see Chapter 7, "Determining Test Coverage."

Tasks

🐦 How to Manage Component Groups

This task describes how to create and manage component groups. It also describes how to associate components to a group and how to check the testing coverage for the entire group.

This task includes the following steps:

➤ "Create an application component group" on page 101

➤ "Add a component to a group" on page 101

➤ "View and Modify Group Details - optional" on page 102

➤ "Mail Group Information - Optional" on page 102

➤ "Results and Coverage" on page 102

Create an application component group

Create a group at the root level of the component or under a specific folder. ALM displays the groups in the component tree hierarchy. For details, see "New Group Dialog Box" on page 103.

Add a component to a group

1 Select a group in the application components tree.

2 Click the **Modeling** tab.

3 Click **Select** in the **Modeling** tab to open the application component folders in the right pane.

4 Expand the tree and select the application component you want to add to the group.

5 Click **Add** > **Associate** to add the component to the group.

6 Drag the entities in the **Modeling** tab into the desired positions, or click the **Arrange Layout** button to do this automatically.

Tip: If you cannot see the group layout, try refreshing the window or expanding it with the splitter bar.

For more details, see the "Modeling Tab" on page 43.

View and Modify Group Details - optional

Select a group in the application components tree and click the **Details** tab.

For further information, see the "Details Tab for Groups" on page 37.

Mail Group Information - Optional

Send group details by e-mail to other users. This enables you to routinely inform users about the status of a group and its components. The e-mail message includes a link that enables the recipient to go directly to the group.

For more information about sending e-mails directly from ALM, see the *ALM User Guide*.

Results and Coverage

Click the **Testing Status** tab to view the complete coverage for **Aspects**, **Requirements**, or **Tests.**

For details, see Chapter 7, "Determining Test Coverage."

Reference
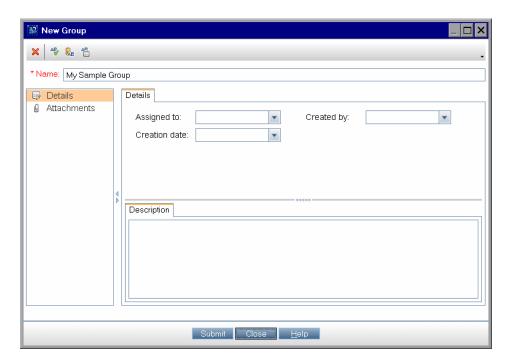
🔍 New Groups User Interface

This section includes:

➤ New Group Dialog Box on page 103

🔍 New Group Dialog Box

This dialog box enables you to create a new application component group.

To access	Use one of the following: ➤ Select **Components** > **New** > **New Group.** ➤ Select **New** > **New Group** from the shortcut menu.
Relevant tasks	"Create an application component group" on page 101

The following elements are included:

UI Elements (A-Z)	Description
✖	**Clear All Fields.** Clears all the fields in the dialog box so that you can redefine the group.
📝	**Details module.** Enables access to the **Details** and **Description** tabs.
📎	**Attachments module**. Allows you to attach a file to the group. For details, see the *ALM User Guide*.
Submit	**Submit.** Creates the group whose details you specified.
Close	**Close.** Closes the dialog box and cancels any changes you made to the existing screen.
Description tab	A description of the group's objective.
Details tab	Lets you set the following fields: ➤ **Assigned To** ➤ **Created By** ➤ **Creation Date**
Name	The name of the group. This field is required.

6

Generating Requirements and Tests

This chapter includes:

Concepts

➤ Generating Requirements and Tests Overview on page 106

Tasks

➤ How to Generate Requirements and Tests on page 110

Reference

➤ Generate Requirements and Tests User Interface on page 112

Concepts

🍀 Generating Requirements and Tests Overview

To test your application, you can manually create requirements and tests, or use the Requirement and Test Generation wizard to automatically generate requirements and test scripts.

The Requirement and Test Generation wizard guides you through the process of creating requirements and scripts to test your application components. Through the wizard, you indicate which aspects of the component you want to test. These aspects include interoperability with different toolkits, boundary testing, and standard compliance.

After you select the testing aspects, Service Test Management generates a set of requirements, or requirements and tests linked to the component or to components in the group. Only requirements are linked to components. The tests are indirectly linked to components through requirements. These requirements and tests provide pre-packaged testing methodologies and component information that specify which components of the application need to be tested, and what to test in each part.

Generated requirements provide a textual description of the testable feature and the expected behavior, based on a template of the specified testing aspects adapted to the selected components. Generated tests describe the test objective and provide an implementation of that testing objective. They include the test operations and arguments for specific testing types. After generating requirements and tests, you should customize them to your particular requirements.

You can also link components to manually created requirements, and indirectly to tests through requirements. For details on assigning components to requirements and tests, see "Linking Components and Requirements" on page 147.

This section also includes:

➤ "Testing Aspects" on page 107

➤ "Actions after Creating a Test" on page 109

Testing Aspects

The Requirement and Tests Generation wizard helps you create requirements and tests that verify different aspects of your component. Service Test Management creates a separate requirement for each aspect and child aspect, and a separate test for each child aspect.

The following table lists the built-in testing aspects:

Aspect Name	Description
Positive Testing	Generates a full positive test that checks each operation of the component. The wizard creates a separate **Positive Testing** test for each of the service's operations.
Standard Compliance	Checks the component's compliancy with industry standards such as WS-I and SOAP.
Service Interoperability	Tests the interoperability of the component's operations with all supported technological platforms. Tests that the component is fully interoperable with all supported technological platforms, by calling all of its operations with default or expected values. For Web Services, it contains the following sub-aspects: ➤ **.NET Framework.** Tests that the service is fully interoperable with .NET Framework WSE 2 toolkit. ➤ **Axis/Java Based Web Services.** Tests that the service is fully interoperable with Axis 1.3 Web Services.

Aspect Name	Description
Security Testing	Tests security. Contains the following sub-aspects: ➤ **SQL Injection Vulnerability.** Checks if the component is vulnerable to SQL injections by injecting SQL statements and errors into relevant parameters. ➤ **Cross-site Scripting (XSS).** Attempts to hack the component by injecting code with relevant parameters that will disrupt its functionality.
Boundary Testing	Using the negative testing technique, creates tests to manipulate data, types, parameters, and the actual SOAP message to test the component to its limits. Contains the following sub-aspects: ➤ **Extreme Values.** Provides invalid data types to the components and verifies they are not accepted. ➤ **Null Values.** Provides NULL parameters to the components to verify they are not accepted.
Performance Testing	Contains the following sub-aspects: ➤ **Stress Testing.** Tests the maximum load that can be placed on the application. ➤ **Overload Sustainability Testing.** Tests how well the hardware allocated for the application can support the number of anticipated users. ➤ **Volume Testing.** Tests that the system can handle a massive data entry. ➤ **Longevity Test.** Tests that the system can sustain a consistent number of concurrent users and executing transactions using near-peak capacity, over a minimum 24-hour period. ➤ **Scalability Testing.** Repeated stress, overload, volume, and longevity tests with different server or network hardware configurations.

You can also define custom aspects. For details, see the Customization's "Aspects Tab" on page 179.

Actions after Creating a Test

After generating requirements or requirements and tests, you can do the following:

Action	Description
Edit the generated requirements	You can modify the details, attachments, component coverage, test coverage, and defect links for any requirement to suit your specific testing objectives. For example, you might want to define customized thresholds that are not reflected in your component definitions. For details about defining requirements, refer to the *HP ALM User Guide*.
Edit the generated tests	You can edit the generated scripts to expand their scope and purpose. You can edit manual tests, scripts designed with Service Test, or those created by a third-party vendor.
Configure run-time settings for test instances	You can view the script's run-time settings, and configure run-time settings for test instances in the **Test Lab** module. For details, refer to the *HP ALM User Guide*.
Configure parameters for test instances	You can configure parameters for test instances in the **Test Lab** module. For details, refer to the *HP ALM User Guide*.
Manually set requirements	You can manually set requirements for a component. For details on linking components to requirements, see "Linking Components and Requirements" on page 147.
Manually add test coverage	You can manually add test coverage for a component. For details, see "Linking Components and Requirements" on page 147.
Run the test	Run the test from ALM's **Test Lab** module. For details, see the *ALM User Guide*.

Tasks

🔧 How to Generate Requirements and Tests

This task describes how to create requirements or tests for checking the quality of your component.

This task includes the following steps:

➤ "Prerequisites" on page 111

➤ "Run the wizard - optional" on page 111

➤ "Review the generated tests" on page 111

➤ "Remove unwanted tests or requirements - optional" on page 111

1 **Prerequisites**

Make sure you have at least one component in the application components tree. If not, use the **Components** menu to create one. Select the component for which you want to create requirements or tests.

2 **Run the wizard - optional**

To create aspect-based tests using a wizard, select **Components** > **Generate Requirements/Tests**. For details, see the "Generate Requirements and Tests Wizard" on page 112.

3 **Review the generated tests**

After the generation process, you can review the generated entities in the **Testing Status** tab. This tab also provides full coverage information for each requirement and test. For details, see Chapter 7, "Determining Test Coverage."

4 **Remove unwanted tests or requirements - optional**

To remove a test or requirement, select it in the **Testing Status** tab and click the **Delete** button. For details, see Chapter 7, "Determining Test Coverage."

Reference

🔍 Generate Requirements and Tests User Interface

This section includes:

➤ Generate Requirements and Tests Wizard on page 112

🔍 Generate Requirements and Tests Wizard

This wizard enables you to create requirements and/or tests for your component.

To access	Select **Components** > **Generate Requirements/Tests.**
Relevant tasks	"Generate Requirements and Tests Wizard" on page 112
Wizard map	This wizard contains: Select Entities to Generate Page > Select Testing Aspects Page > Select Locations Page > Summary Page > Generate Page
See also	➤ "View the Requirements coverage - optional" on page 128 ➤ "View the Tests coverage - optional" on page 129

🔍 **Select Entities to Generate Page**

This wizard page enables you to indicate which entities to create: requirements and tests, tests only, or requirements only.

Important information	General information about this wizard is available here: "Generate Requirements and Tests Wizard" on page 112.
Wizard map	This wizard contains: **Select Entities Page** > Select Testing Aspects Page > Select Locations Page > Summary Page> Generate Page

User interface elements are described below:

UI Elements (A-Z)	Description
Generate	The entities to generate: ➤ **Requirements and tests.** Generates a set of requirements, and tests that are indirectly linked to the component through the requirements. The tests describe the test objective, based on a template of the specified testing aspects. Whenever possible, generated tests provide an implementation of the test objective by describing the test operations. ➤ **Tests only.** Generates a set of tests for the aspects you select in the next step. The tests are indirectly linked to the component via the component's default requirement. ➤ **Requirements only.** Generates a set of requirements for the selected component.
Select Script Generator	The engine to use for generating a test. ➤ **Manual Test.** Creates manual tests with no design steps. This is available for all application component types. ➤ **ST11 Generator.** Creates Service Test tests (only available for Web Service type components when Service Test version 11.10 or higher is installed). This generator is only available for Web Services with document/literal type WSDLs that can be parsed by Service Test.

🔍 Select Testing Aspects Page

This wizard page enables you to select testing aspects for which to create the
requirement and tests.

Important information	General information about this wizard is available here: "Generate Requirements and Tests Wizard" on page 112.
Wizard map	This wizard contains: Select Entities to Generate Page > **Select Testing Aspects Page** > Select Locations Page > Summary Page > Generate Page
See also	"Customizing Service Test Management" on page 161

User interface elements are described below:

UI Elements (A-Z)	Description
☐ 🔊	**Clear aspect.** Excludes the aspect.
☑ 🔊	**Select Aspect.** Includes the aspect. If you select a parent, all of its children are included.
Covered aspects	A tree view of all the aspects available for your components. Use the **Expand** ⊞ and **Collapse** ⊟ buttons to show or hide all of the aspects.

🔍 Select Locations Page

This wizard page enables you to indicate which entries to create: requirements and tests, tests only, or requirements only.

Important information	General information about this wizard is available here: "Generate Requirements and Tests Wizard" on page 112.
Wizard map	This wizard contains:
	Select Entities to Generate Page > Select Testing Aspects Page > **Select Locations Page** > Summary Page > Generate Page

User interface elements are described below:

UI Elements (A-Z)	Description
[...]	**Browse.** Locates a parent requirement or test folder.
<parent requirement>	Generate requirements as children of the selected requirement.
<tests folder>	Generate tests under the selected test folder.

🔍 Summary Page

This wizard page provides a summary of the tests and requirements that are to be generated.

Important information	When you click **Next** in this screen, the wizard begins generating the requirements and tests. You cannot stop generation in the middle.
Wizard map	This wizard contains:
	Select Entities to Generate Page > Select Testing Aspects Page > Select Locations Page > **Summary Page** > Generate Page

The summary sections are described below:

UI Elements (A-Z)	Description
Covered aspects	A list of the aspects you selected in the **Testing Aspects** screen.
Generated requirements will be stored as children of…	The parent requirement under which the new requirements will be generated.
Test destinations	The folder under which the tests will be generated.

Generate Page

This wizard page enables you to select locations to store the generated entities: a parent requirement (for generated requirements) and a test folder (for generated tests).

Important information	General information about this wizard is available here: "Generate Requirements and Tests Wizard" on page 112.
Wizard map	This wizard contains: Select Entities to Generate Page > Select Testing Aspects Page > Select Locations Page > Summary Page > **Generate Page**

User interface elements are described below:

UI Elements (A-Z)	Description
Finish	Closes the wizard. This button is disabled until the test and requirement generation is complete.
Generate grid	A grid containing a list of the requirements and tests that were generated. The grid contains the following columns: ➤ **Entity type**: Requirement or Test ➤ **Generation status**: Passed or Failed ➤ **Testing aspect**: The aspects to cover. For details, see "Testing Aspects" on page 107. ➤ **Component**: Component name
Generation Status Details	Provides the reasons for failed generation of requirements or tests.

7

Determining Test Coverage

This chapter includes:

Concepts

Tasks

Reference

Concepts

🔧 Coverage Overview

You analyze the quality of your application components using charts that display the status of the testing elements: aspects, requirements, tests, operations, and defects. You can analyze either a specific component or all the components in a group or folder.

The **Testing Status** tab shows the coverage of the requirements and tests as they were defined for the components.

This section also includes:

➤ "Rule Violations" on page 120

➤ "Aspect Coverage" on page 121

➤ "Requirements Coverage" on page 122

➤ "Operations Coverage" on page 123

➤ "Tests Coverage" on page 124

➤ "Defects Coverage" on page 126

🔧 Rule Violations

The **Testing Status > Rules** tab shows you the rules that were violated by the component.

The following table describes the rule violations.

Requirement Status	Description
Rule name	The name given to the rule.
Rule description	A description of the rule.

For task details, see "View the Rule Violations - optional" on page 129.

🟦 Aspect Coverage

The Testing Status **Aspects** tab lets you see the coverage of the Testing aspects defined for the component testing. This information is dynamic and displays the current testing status in percentages. It is useful for the business analyst or system architect who wants to see the coverage in meaningful terms.

The following table describes the aspect statuses.

Aspect Status (A-Z)	Description
All Failed	All the aspect's requirements are assigned the **Failed** status.
All Passed	All the aspect's requirements are assigned the **Passed** status.
No Run/Not Completed	All the aspect's requirements are assigned either the **No Run, Not Completed** or **Not Covered** status.
Not Covered	All the aspect's requirements are assigned the **Not Covered** status.
Some Failed	At least one of the aspect's requirements is assigned the **Failed** status.
Some Passed	At least one of the aspect's requirements is assigned the **Passed** status. All other requirements have either the **No Run**, **Not Completed**, or **Not Covered** status.

🔹 Requirements Coverage

Requirement coverage is used to connect components to other ALM entities (tests, test sets, and defects) through the requirement. This helps you keep track of the relationship between your requirements and components, and ensures compliance with your requirements throughout the testing process. After you have created tests, you associate assigned requirements with tests and defects. In this way, you can keep track of your testing needs at all stages of the testing process. If a requirement changes, you can immediately identify which tests and defects are affected, and who is responsible.

To test application components in Service Test Management, you must link them to requirements. When you define a component, you can link it to an existing requirement or create new requirements.

To use existing requirements, you drag them from the Requirement entities tree to your component.

To create new requirements, use the Requirement and Test Generation wizard as described in "Generating Requirements and Tests" on page 105.

The **Testing Status** > **Requirements** tab lets you see the coverage of the requirements that were linked to the component. This information is dynamic and displays the current status in percentages. It is most useful for advanced users who understand the ALM coverage model and want to integrate application components into that model.

The following table describes the requirement statuses.

Requirement Status (A-Z)	Description
Failed	At least one test associated with the requirement has failed.
N/A	The terms **Passed** and **Failed** do not apply to the requirement.
No Run	All tests associated with the requirement have a **No Run** status.
Not Completed	All tests associated with the requirement have the **Not Completed** status.
Not Covered	There are no tests associated with the requirement.
Passed	All test associated with the requirement have passed.

To manually link requirements to components, see Chapter 8, "Linking Components and Requirements."

Operations Coverage

The **Testing Status** > **Operations** tab lets you see the testing status per operation in chart form. This is useful when the component is less granular and its operations may be considered as separate use cases for quality assurance.

When you expand an operation, you can see the status of the tests associated with the operation. In this context, association indicates that:

➤ the test is linked to a requirement which is linked to this component

➤ the test uses this operation.

The following table lists the statuses:

Aspect Status (A-Z)	Description
All Failed	All of the component's operations had a **Failed** status for their tests.
All Passed	All of the component's operations had a **Passed** status for their tests.
No Run/Not Completed	All of the component's operations had the status of **No Run, Not Completed** or **Not Covered** for their tests.
Not Covered	All of the component's operations had the status of **Not Covered** for their tests.
Some Failed	At least one of the component's operations had a **Failed** status for their tests.
Some Passed	Some of the component's operations had a **Passed** status for their tests. All other's had either the **No Run**, **Not Completed**, or **Not Covered** status.

Tests Coverage
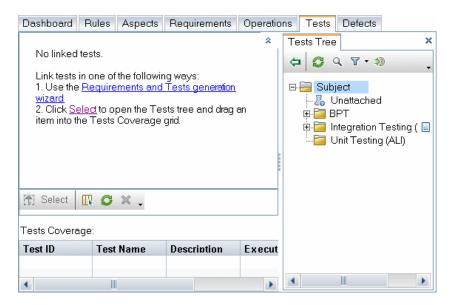
The **Testing Status > Tests** tab lets you see the coverage of the tests that were linked to the component. This information is dynamic and displays the current status in percentages.

The following table lists the possible statuses:

Status (A-Z)	Description
Blocked	The percentage of tests that were blocked from being run.
Failed	The percentage of tests that returned a **Failed** status.
N/A	The percentage of tests to which the terms **Passed** and **Failed** do not apply.

Status (A-Z)	Description
No Run	The percentage of tests that were not run.
Not Completed	The percentage of tests that were not completed.
Passed	The percentage of tests that returned a **Passed** status.

If there are tests linked to the application component, Service Test Management provides links to the **Requirements and Test Generation wizard** or to the **Tests Tree** tab.

Alternatively, you can link components to tests through the **Test Plan** module.

Note: You can create automatic coverage between test instances and components using the **ALLOW_REQ_COVERAGE_BY_TEST_INSTANCE** parameter in Site Administration. For details about this parameter, refer to the *HP Application Lifecycle Management Administrator's Guide*.

🍥 Defects Coverage

The **Testing Status** > **Defects** tab lets you see the coverage of the defects that were linked to the component. This information is dynamic and is displayed in a chart form, based on the severity of the defects.

You do not link a defect directly to a component. Instead, you link a defect to a requirement or test. If the component was assigned a test that is linked to a defect, Service Test Management displays the defect status in this tab.

For details, see the **Test Plan** module documentation in the *ALM User Guide*.

Tasks

🔧 How to Check the Testing Status

This following steps describe how to check the testing status of your component. You check the testing status by viewing the coverage of the aspects, requirements, operations, tests, and defects associated with your component.

This task includes the following steps:

➤ "View the Aspects coverage - optional" on page 127

➤ "View the Requirements coverage - optional" on page 128

➤ "View the Operations coverage - optional" on page 128

➤ "View the Tests coverage - optional" on page 129

➤ "View the Rule Violations - optional" on page 129

➤ "View the Defects coverage - optional" on page 130

To check the testing status, you do not need to check all of the coverage types. The coverages you need to check depends on how you designed your test plan and your area of focus. In one instance you may be interested solely in the testing aspects, while in another scenario, the test requirements may be the most relevant.

View the Aspects coverage - optional

1 Select a component, group, or folder in the tree hierarchy.

2 Select the **Testing Status** tab and click its **Aspects** button. Service Test Management shows a graphic representation of the coverage in the top pane, and a list of the required aspects in the bottom pane. The graph only shows the aspect coverage—not the requirements. For user interface details, see the "Aspects Tab" on page 135.

3 Expand an aspect to see its associated requirements. Click on a requirement to open the item in the **Requirements** module. For details, see the *HP Application Lifecycle Management User Guide*.

View the Requirements coverage - optional

1 Select a component, group, or folder and click the **Testing Status** > **Requirements** tab.

2 Expand a requirement to see its associated tests. Click on a test to open it in the **Test Lab** module. In the expanded grid, you can drag a test from one parent requirement to another.

3 If there are no requirements associated with this component, use the "Generate Requirements and Tests Wizard" on page 112 to create them. To add existing requirements to the coverage, click **Select** to open the **Requirements Tree** tab in the right pane, and drag an item into the Requirements Coverage grid.

4 Double-click on a requirement in the grid to open it in the **Requirements** module. To return to your last view, click on **Application Components** in the left pane.

For details about the Requirements tab, see "Requirements Tab" on page 137.

View the Operations coverage - optional

1 Select a component and click the **Testing Status** > **Operations** tab.

2 Expand an operation to see the status of the tests associated with the operation.

3 To view the test details, click on a test in the grid to open it in the **Test Plan** module. For details about the **Requirements** and **Test Plan** modules, see the *HP Application Lifecycle Management* User Guide.

4 To return to your last view, click the **Application Components** module button in the left pane.

For user interface details, see the "Operations tab" on page 140.

View the Tests coverage - optional

1 Select a component, group, or folder and click the **Testing Status** > **Tests** tab.

2 To add more tests to a component's coverage, click **Select**. Expand the **Tests Tree** in the right pane and select the test you want to add to the coverage. Use the filter button to show only the desired tests.

3 Drag the test to the **Tests Coverage** grid, or click the **Add to Coverage** button.

4 Click on a test in the **Tests Coverage** grid to open it in the **Test Plan** module. To return to your last view, click **Application Components** in the left pane.

Tip: In the grid, use the right-click menu to create new tests or navigate to an existing test.

For user interface details, see the "Tests Tab" on page 142.

View the Rule Violations - optional

1 Select a component in the application components tree.

2 Select the **Testing Status** > **Rules** tab. Service Test Management shows a list of the rule violations for the selected component.

3 To resolve the violations, click **Resolve All** to open the "Generate Requirements and Tests Wizard". For details see page 112.

For user interface details, see the "Rules Tab" on page 133.

View the Defects coverage - optional

1 Select a component or group in the tree hierarchy.

2 Select the **Testing Status** > **Defects** tab. Service Test Management shows a graphic representation of the coverage in the top pane, and a list of the defects in the bottom pane.

3 Click on a defect to view its details in the **Defects** module. To return to your last view, click the **Application Components** module in the left pane.

4 To refresh the chart, click the **Refresh** button.

For user interface details, see the "Defects Tab" on page 145.

For more information about the **Requirements**, **Test Plan**, and **Defects** modules, see the *HP Application Lifecycle Management* User Guide.

Reference

🔍 Testing Status Tab

This section describes the interface of the Testing Status sub tabs.

🔍 Testing Status Dashboard

The Testing Status dashboard displays a coverage summary of each of the following items: **Aspects**, **Requirements**, **Operations**, **Tests**, and **Defects**.

To access	Do the following:
	➤ Select the **Application Components** module.
	➤ Select an entity in the application components tree.
	➤ Select the **Testing Status** > **Dashboard** tab.
Important information	Click on a graph to open it in a separate tab.
Relevant tasks	"How to Check the Testing Status" on page 127

The graphs are described below. Click on a graph to open it in its own tab.

UI Elements (A-Z)	Description
Aspects	A graphical representation of the coverage by the testing aspects.
Defects	A graphical representation of the coverage for the defects associated with the component.
Operations	A graphical representation of the coverage by the component's operations.
Requirements	A graphical representation of the status of the requirements associated with the component.
Tests	A graphical representation of the status of the tests associated with the component.

Rules Tab

This tab enables you to view and manage the rules violated by the selected component.
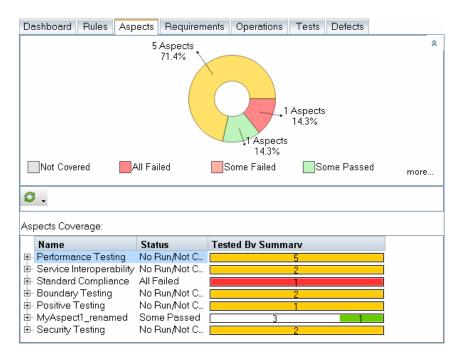
To access	Do the following:
	➤ Select the **Application Components** module.
	➤ Select an application component.
	➤ Select the **Testing Status** > **Rules** tab.
Important information	See "Rule Violations" on page 120.
	Note: Available only for Application Component type entities.
Relevant tasks	"View the Rule Violations - optional" on page 129

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
	Select Columns. Enables you to select the columns to display in the Rule Violations grid.
	Refresh. Reloads the list of rule violations.
	Set Filter/Sort. Enables you to set a filter for the rule violations or sort them by a criteria.
Resolve All	**Resolve All.** Opens the **Generate Requirements and Tests** wizard to resolve the rule violations. For details, see the "Generate Requirements and Tests Wizard" on page 112.
<Rule Violations grid>	A list of all the rule violations for the selected component.
	Default: Rule Name and Rule Description columns.
	Available Columns: Alert ID, Alert is active, and Component name.

✎ Aspects Tab

This tab displays the coverage per testing aspect in both a graphical representation and a grid layout.

To access	Do the following: ➤ Select the **Application Components** module. ➤ Select an entity in the application components tree. ➤ Select the **Testing Status** > **Aspects** tab.
Important information	See "Aspect Coverage" on page 121.
Relevant tasks	"View the Aspects coverage - optional" on page 127
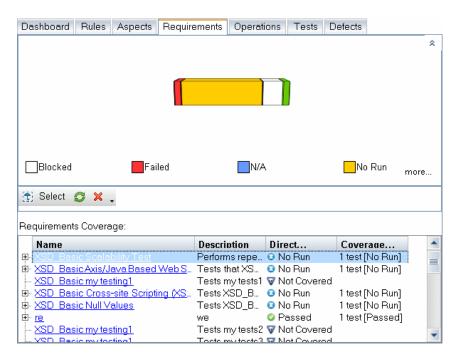
The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
	Refresh. Reloads the status of the testing aspects and refreshes the graph.
⊟	**Collapse.** Shows the aspect only, but not its tests or requirements.
⊞	**Expand.** Shows the tests associated with the aspect.
<graph pane>	A graphical representation of the requirement coverage
<grid pane>	A list of all aspects associated with the component. ➤ Expand the aspect to see a list of its associated tests. ➤ Click on a test to open it in the **Test Lab** module. ➤ Click **Application Components** in the left pane to return to the most recent view of the component.

Requirements Tab

This tab displays the coverage per testing requirement in both a graphical representation and a grid layout.


To access	Do the following: ➤ Select the **Application Components** module. ➤ Select an entity in the application components tree. ➤ Select the **Testing Status** > **Requirements** tab.
Important information	See "Requirements Coverage" on page 122.
Relevant tasks	"View the Requirements coverage - optional" on page 128

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
�The Select	**Select.** Opens the Requirements tree enabling you to add more requirements to your test. For details see "Requirements Tree Panel" on page 138.
🔄	**Refresh.** Reloads the status of the requirements and refreshes the graph.
✖	**Remove.** Unlinks the requirement and the component. Note that unlinking a requirement may remove the linkage to all tests and defects that are associated with the component only through this requirement.
<graph pane>	A graphical representation of the requirement coverage.
<grid pane>	A list of all requirements associated with the component. ➤ Expand a requirement to see a list of its associated tests ➤ Click on a requirement to open it in the **Requirements** module. ➤ Click on a test to open it in the **Test Lab** module. ➤ Click **Application Components** in the left pane to return to the most recent view of the component.

Requirements Tree Panel
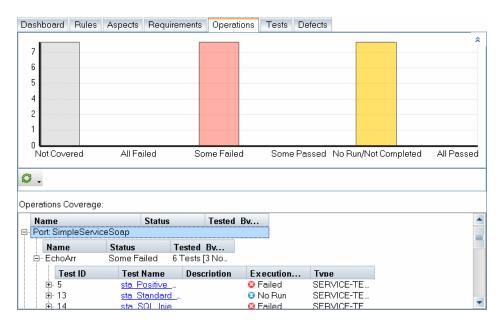
The following elements are included (unlabeled GUI elements are shown in angle brackets):
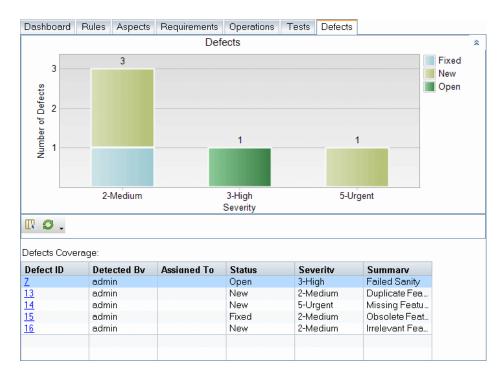
UI Elements	Description
⬅	**Add Coverage.** Adds coverage for the selected requirement. If you select a parent requirement, it adds all sub-requirements.
🔄	**Refresh.** Refreshes the Requirements tree.

UI Elements	Description
🔍	**Find.** Opens the Find dialog box for locating a requirement.
▼ ▾	**Set/Clear Filter.** Sets or clears the filter. ➤ **Set Filter / Sort.** Opens the Set Filter/Sort dialog box allowing you to filter out and sort the requirements by any field. ➤ **Clear Filter / Sort.** Clears all filters and sorting settings and displays all requirements.
⇥	**Go to Requirement by ID**. Locates the requirement whose ID you specify.
✕	**Close.** Closes the Requirements tree pane. To reopen this view, click the **Select** button in the Requirements tab.
<Requirements tree>	A tree view of all of the requirements by folder and group. An icon indicates the requirement type. **Tip:** If a filter is active, there could be hidden requirements. To see all requirements, click **Clear Filter**.

🔖 Operations tab

This dialog box displays the coverage per operation in both a graphical representation and a grid layout.
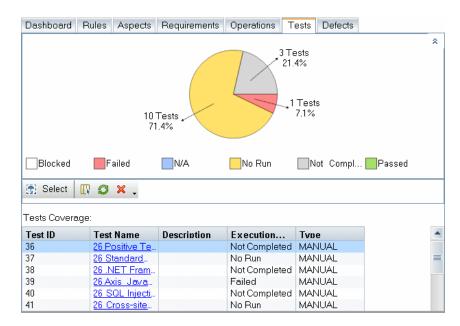
To access	Do the following:
	➤ Select the **Application Components** module.
	➤ Select an entity in the application components tree.
	➤ Select the **Testing Status** > **Operations** tab.
Important information	See "Operations Coverage" on page 123.
	Note: Available only for Application Component type entities.
Relevant tasks	"View the Operations coverage - optional" on page 128

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
(refresh icon)	**Refresh.** Reloads the status of the operations and refreshes the graph.
(collapse icon)	**Collapse.** If there are multiple ports, collapsing the node shows the port name, while hiding the operations.
(expand icon)	**Expand.** Shows the operations and their testing status.
<graph pane>	A graphical representation of the Operations coverage.
<grid pane>	A list of all operations associated with the component. ➤ Expand the operation to see a list of its associated tests. ➤ Click a test to open it in the **Test Lab** module. ➤ Click **Application Components** in the left pane to return to the most recent view of the component.

🔍 **Tests Tab**

This dialog box displays the status per test in both a graphical representation and a grid layout.

To access	Do the following: ➤ Select the **Application Components** module. ➤ Select an entity in the application components tree. ➤ Select the **Testing Status** > **Tests** tab.
Important information	See "Tests Coverage" on page 124.
Relevant tasks	"View the Tests coverage - optional" on page 129

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
🔲 Select	**Select.** Opens the Tests tree enabling you to associate more tests with your component. For details see "Tests Tree Panel" on page 143.
🔳	**Select Columns.** Opens the Select Column dialog box enabling you to select which columns to show in the grid.
🔄	**Refresh.** Reloads the status of the test and refreshes the graph.
✖	**Remove.** Unlinks the test from the requirement through which it is associated with the component.
<graph pane>	A graphical representation of the test coverage.
<grid pane>	A list of all tests associated with the component. ➤ Click test link to go to the test in the **Test Lab** module. ➤ Click the **Application Component**s to return to the most recent view of the component.

Tests Tree Panel

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
⬅	**Add Coverage.** Adds the selected test to the coverage of the component or requirement. If you select a folder or group, it adds all sub tests.
🔄	**Refresh.** Refreshes the Tests tree.
🔍	**Find.** Opens the Find dialog box for locating a requirement.

🔍 Defects Tab

This dialog box displays the status of the defects in both a graphical representation and a grid layout. It shows their status and the number of the defects, and provides direct links to those defects.

To access	Do the following:
	➤ Select the **Application Components** module.
	➤ Select an entity in the application components tree.
	➤ Select the **Testing Status** > **Defects** tab.
Important information	See "Defects Coverage" on page 126.
	Note: Available only for Application Component and Group type entities.
Relevant tasks	"View the Defects coverage - optional" on page 130

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
	Select Columns. Opens the Select Column dialog box enabling you to select which columns to show in the grid.
	Refresh. Reloads the status of the defects and refresh the graph.
<graph pane>	A graphical representation of the defects coverage.
<grid pane>	A list of all defects associated with the component and their status. ➤ Click on a defect in the **Defect ID** column to open it in the **Defects** module. ➤ Click **Application Components** in the left pane to return to the most recent view of the defects coverage.

8

Linking Components and Requirements

This chapter includes:

Concepts

➤ Linking Components Overview on page 148

Tasks

➤ How to Link Components to Requirements and Tests on page 149

Reference

➤ Linking Component User Interface on page 151

Concepts

🍥 Linking Components Overview

To test components in Service Test Management, you must link them to requirements or tests. Requirement coverage is used to connect components to other ALM entities such as tests, test sets, and defects.

Linking components to requirements and tests helps you keep track of the relationship between your requirements and components, and ensures compliance with your requirements throughout the testing process.

For information on viewing test and requirement coverage, see Chapter 7, "Determining Test Coverage."

When working in the **Requirements** module, you can view the application components linked to each requirements. The **Application Components** tab displays a list of those components, with other basic information.

Using the **Test Lab** module's **Application Components** tab (available only after you install the Service Test Management extension), you can set and view the application components linked to each test.

When you link a component to a test, Service Test Management prompts you to select a requirement. To allow you to track your testing through ALM, you link tests to the default requirement or to an existing one.

Tasks

🦴 How to Link Components to Requirements and Tests

This task describes how to link components to either requirement or tests. If you are linking to a test, you also indicate the requirement to associate with the test.

This task includes the following steps:

➤ "Link a component to a requirement - optional" on page 149

➤ "Link a component to a test - optional" on page 149

➤ "Results" on page 150

Link a component to a requirement - optional

Use the **Application Components** tab in the **Requirements** module to link components to requirements. For details, see "Application Components Tab - Requirements Module" on page 154.

Link a component to a test - optional

Use the **Application Components** tab in the **Test Plan** module to link a components to tests. For details, see "Application Components Tab - Test Plan Module" on page 156.

When you add a test to component coverage, the Link <test-name> to Requirement dialog box opens and prompts you to choose the **Default Requirement** or **Linked Requirements**. Select a requirement and click **OK**. For details, see the "Link <test-name> to Requirement Dialog Box" on page 152.

You can also link a component to a test through the **Testing Status** tab. For details, see "Tests Coverage" on page 124.

Results

You can view and modify the linkages in the **Application Components** tab within the **Test Plan** or **Requirements** modules. For details, see "Linking Component User Interface" on page 151.

Reference
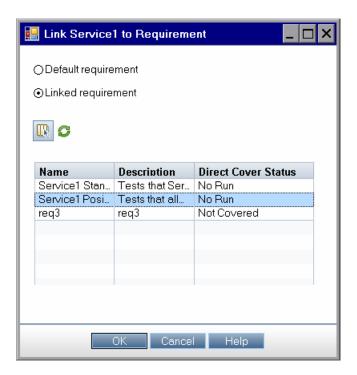
🔍 Linking Component User Interface

This section includes (in alphabetical order):

➤ Link <test-name> to Requirement Dialog Box on page 152

➤ Application Components Tab - Requirements Module on page 154

➤ Application Components Tab - Test Plan Module on page 156

➤ Application Components Tree Panel on page 158

🔍 Link <test-name> to Requirement Dialog Box

This dialog box enables you to select a requirement to associate with your test.
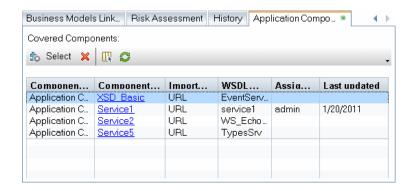
To access	Do the following:
	1 Open the **Test Plan** module, and select the **Application Components** tab (you may need to scroll to the right).
	2 Click **Select** to open the selection panel, and choose a entity from the **Application Components Tree** tab.
	3 Click the ⬅ button.
Important information	This dialog box only opens if the application component is linked to a requirement other than the default one.
Relevant tasks	"Link a component to a test - optional" on page 149
See also	"General Tab" on page 172 (Customization)

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements (A-Z)	Description
	Select Columns. Opens the Select Column dialog box enabling you to select the columns to display in the grid.
	Refresh. Reloads the components in the list.
<Linked Requirements list>	A list of the requirements associated with the test. To modify the columns, click the **Select Columns** button.
Default requirement	**Default Requirement.** Creates a link between the default requirement of the component and the test, if it does not already exist. The linked requirements list is not relevant for this option. You can set the default requirement in the Customization section, For details, see "Select Requirement" on page 174.
Linked requirement	Enables you to select a specific a requirement to link to the test. Select a requirement from the Linked Requirements list. This option is not available if the component or group has no linked requirements. To link a requirement with a component, see "View the Requirements coverage - optional" on page 128.

✏️ Application Components Tab - Requirements Module

This dialog box enables you to link components to an existing requirement.

To access	Do the following:
	➤ Open the **Requirements** module.
	➤ Select **View** > **Requirement Details**.
	➤ Expand the tree and select a requirement.
	➤ In the right pane, scroll right to **Application Components** tab.
Relevant tasks	"How to Link Components to Requirements and Tests" on page 149
See also	"How to Check the Testing Status" on page 127

User interface elements are described below (unlabeled elements are shown in angle brackets):
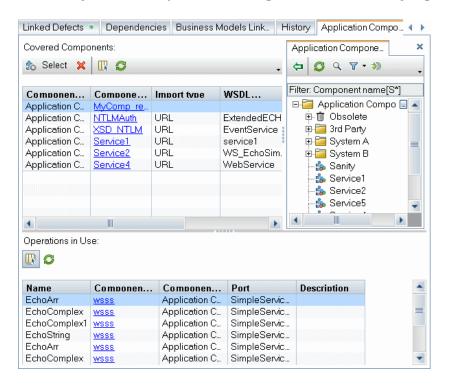
UI Elements (A-Z)	Description
🔩 Select	Opens the **Application Components Tree** tab in the right pane to link a component to the requirement. For details, see the "Application Components Tree Panel" on page 158.
✖	**Delete.** Unlinks the component and requirement.

UI Elements (A-Z)	Description
	Select Columns. Opens the Select Column dialog box enabling you to select the columns to display in the grid.
	Refresh. Reloads the components in the grid.
<Linked Components Grid>	A list of the components linked to the selected requirement. The displayed columns differ per component type. For Web Services, they are: ➤ **Component folder path.** The application component path in the tree hierarchy. ➤ **Component name.** The name of the application component. ➤ **Import type.** For Web Services, the import type: File, URL, UDDI, Systinet ➤ **WSDL native name.** The Web service name as defined in the WSDL. ➤ **Assigned to QA.** QA engineer responsible for validating the requirement. ➤ **Last updated.** The date the component was last updated from its original source. To change the columns displayed, click the button.

🔍 Application Components Tab - Test Plan Module

This tab enables you to link components to an existing test.

This dialog box enables you to link components to an existing requirement.

To access	Do the following:
	➤ Open the **Test Plan** Module.
	➤ Select **View** > **Test Plan Tree**.
	➤ Expand tree and select a test in the left pane.
	➤ In the right pane, scroll right to the **Application Components** tab.
Relevant tasks	"How to Link Components to Requirements and Tests" on page 149
See also	"How to Check the Testing Status" on page 127

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements (A-Z)	Description
🔧 Select	Opens the **Application Components Tree** panel in the right pane to enable linking a component to the test. For more information, see the "Application Components Tree Panel" on page 158.
✖	**Delete.** Unlinks the test from all requirements that associate it with the component.
🔲	**Select Columns.** Opens the Select Column dialog box enabling you to select the columns to display in the grid.
🔄	**Refresh.** Reloads the list in the grid.
<**Linked Components grid**>	A list of components linked to the test. The displayed columns may differ based on the component type. ➤ **Component folder path.** The path in the tree hierarchy. ➤ **Component name.** The name of the application component. ➤ **Import type.** The import type: File, URL, UDDI, Systinet (Web Services only). ➤ **WSDL native name.** The service name as defined in the WSDL (Web Services only). ➤ **Assigned to QA.** QA engineer responsible for validating the test. ➤ **Last updated.** The date the component was last updated from its original source. To change the columns displayed, click the 🔲 button.

Application Components Tree Panel

This tab enables you to link components to an existing test.

To access	Do the following:
	4 Open the **Test Plan** or **Requirements** module.
	5 Select **View** > **Test Plan Tree** or **View** > **Requirement Details**.
	6 In the right pane, scroll right to the **Application Components** tab.
	7 Click the **Select** button.
Relevant tasks	"How to Link Components to Requirements and Tests" on page 149
See also	"How to Check the Testing Status" on page 127

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
	Add Covered Component. Adds the selected component to the Linked Components grid.
	Refresh. Refreshes the **Application Components Tree** panel.
	Find. Opens the Find dialog box for locating an application component by its name or another criteria. For details, click F1 in the Find dialog box.
	Set/Clear Filter. Opens a drop-down list with the following options:
	➤ **Set Filter / Sort.** Opens the Set Filter/Sort dialog box allowing you to filter and sort the components.
	➤ **Clear Filter / Sort.** Clears all filters and sorting settings and displays all components and groups.
	Go to Application Component by ID. Locates the application component whose ID you specify.

UI Elements	Description
✖	Closes the **Application Components Tree** panel. To reopen this view, click the **Select** button.
<Application Components panel>	A tree view of all of the application components by folder and group (only visible when you click **Select**). **Tip:** If a filter is active, there could be hidden components. To see all components, click **Clear Filter**.

9

Customizing Service Test Management

This chapter includes:

Concepts

➤ Customization Overview on page 162

Tasks

➤ How to Customize a Project on page 167

Reference

➤ Project Customization User Interface on page 171

Concepts

🔩 Customization Overview

You can customize general project data in the **Application Components** module. This includes specifying default values for components stored in secure locations and on a proxy server, the default Systinet and UDDI server information, the toolkit parsing order for importing component definitions, the default requirement folder, user-defined testing aspects, component types, and rules.

You can also customize component, group, and change entities the same way you can customize other ALM entities. You can restrict which user groups can add, delete and modify folders, components, and groups. You can determine whether user groups can update components and groups, and link components and groups to requirements.

In addition, you can customize access to the **Application Components** module for each user group.

Note: This chapter describes the additional customization options available with Service Test Management. For details on Project Customization, refer to the *HP Application Management Lifecycle Administrator's Guide*.

This section also includes:

➤ "Testing Aspects" on page 163

➤ "Rules" on page 164

➤ "Custom Component Types" on page 164

➤ "Application Component Permissions" on page 165

➤ "Application Components Module Access" on page 165

➤ "Custom Project Entities" on page 165
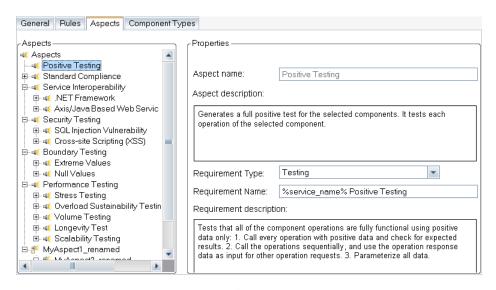
Testing Aspects

Testing aspects are criteria you choose for testing an application component. You can indicate the testing aspects to which the component must comply in order it to receive a Passed status in testing. For example, you can specify Positive Testing and Standard Compliance. For a list of the available aspects, see "Testing Aspects" on page 107.

When using the **Requirement and Test Generation** wizard (**Components > Generate Requirements/Tests)**, you select the aspects you want to test and the wizard creates all of the necessary tests and requirements. For details about the wizard, see the "Generate Requirements and Tests Wizard" on page 112.

Service Test Management's customization section lets you control which aspects to display in the wizard.

The customization's **Aspects** tab also lets you add new aspects, modify the properties of system and user-defined aspects, and delete user-defined aspects.

In the Aspects tree hierarchy, the ◀◎ icon indicates a system-defined aspect. The ◀◎ icon indicates a user-defined aspect.

In the **Rules** tab, you select the aspects for user-defined rules. Application components that were not tested for these aspects, generate a rule violation and appear with an alert in the application component tree. For details, see the "Rules Tab" on page 175.

For task details, see "Customize the aspects - optional" on page 168.

Rules

Rules let you define the testing aspects to which your component must comply. They help you make sure your testing standards are satisfactory and that the aspects relevant to the component are covered.

You define a rule through filters and aspects:

➤ **Filters.** Filters indicate the components to which the rule will be applied. For example, if you indicate a filter: **Assigned To=john_qa**, any component not assigned to **john_qa**, will not be bound by the rule. You can set a rule for a component based on a variety of filters. You can use the conditional and logical operators that are used in all ALM filters. For details, see the *HP Application Lifecycle Management User Guide*.

➤ **Aspects.** The testing aspects that must be covered for the application component.

If your application component was not covered by the required aspects, Service Test Management creates an **Alert**. The **Testing Status > Rules** tab lists the violations, and lets you resolve the problems. For details about defining rules, see "Customize the component rules - optional" on page 168.

Custom Component Types

The customization lets you define new component types. The built-in types are General, Web Service, and JMS.

You can specify the following information for your component: **Name**, **Assigned Icon**, **Supported file extensions**, **Contract parser ID**, and the fields to display in its **Details** and **Interaction** tabs. The **Contract parser ID** instructs Service Test Management that if a a parser has this unique ID and is registered on the client machine, use it to parse contract files of this type.

For details, see "Manage the component types - optional" on page 169.

Application Component Permissions

You can customize the permission settings for user groups and folders and the ability to link relating to their access to application components, linkage to requirements, and ability to associate application components with groups. For details about setting permissions, see "Customize group permissions - optional" on page 169.

For more information about user group permissions, refer to the *HP Application Management Lifecycle Administrator's Guide*.

Application Components Module Access

After you deploy HP Service Test Management for a project, you can control access to the **Application Components** module for each user group. You can enable access to all ALM modules including the **Application Components** module, or to all ALM modules excluding the **Application Components** module.
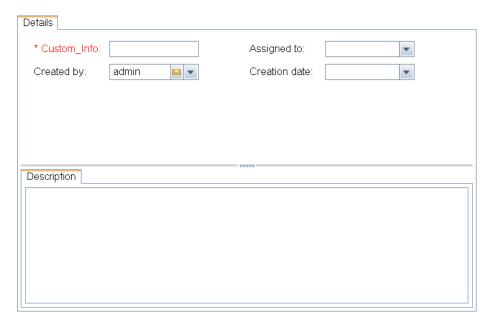
For details about customizing module access for user groups, refer to the *HP Application Management Lifecycle Administrator's Guide*.

Custom Project Entities

You can customize the system fields and create new user-defined fields for Application Components, Application Component Changes, and Application Component groups. You can create new fields and indicate whether or not a field is mandatory.

If you define a field as required, the **Details** tab shows it as mandatory for all component types. The following example shows a mandatory custom field defined for an Application Component group.

Details

* Custom_Info: Assigned to:

Created by: admin   Creation date:

Description

For task details about customizing project entities, see "Customize project entities - optional" on page 170 and the *HP Application Management Lifecycle Administrator's Guide*. For more information on ALM entities and fields, refer to the *HP ALM Database Reference*.

Tasks

🔧 How to Customize a Project

This task describes how to customize your ALM project as it relates to components. You can create rules, assign aspects, and set permissions for entities related to your application component. You can also create both mandatory or optional fields to the Component Details.

All of the customization settings are optional.

This task includes the following steps:

- ➤ "Open Project Customization" on page 167
- ➤ "Customize the General settings - optional" on page 168
- ➤ "Customize the component rules - optional" on page 168
- ➤ "Customize the aspects - optional" on page 168
- ➤ "Manage the component types - optional" on page 169
- ➤ "Customize group permissions - optional" on page 169
- ➤ "Customize application component module access - optional" on page 170
- ➤ "Customize project entities - optional" on page 170

Open Project Customization

- ➤ Choose **Tools** > **Customize** to open the Customization module.
- ➤ Click the **Return** button, located in the top right corner, to return to the main ALM window.

Customize the General settings - optional

Click **Application Components** in the left pane. Select the **General** tab. Define the parsing order for toolkits, and the default servers that host the WSDL, UDDI registry, or Systinet registry. You can also define the authentication details for accessing WSDL files, the default requirement and whether to allow fuzzy imports.

For user interface details, see the "General Tab" on page 172.

Customize the component rules - optional

Click **Application Components** in the left pane. Select the **Rules** tab.

Click **New** to define new rules. Select the testing aspects to which the components in the current filter, must comply. If the component is not covered by the specified aspects, it issues a rule violation. For details, see "Rule Violations" on page 120.

For user interface details, see the "Rules Tab" on page 175.

Customize the aspects - optional

Click **Application Components** in the left pane. Select the **Aspects** tab. Manage the built-in testing aspects or click **New** to create new ones.

For user interface details, see the "Aspects Tab" on page 179.

Manage the component types - optional

Click **Application Components** in the left pane**.** Select the **Component Types** tab.

Manage the built-in component types **General**, **Web Services**, and **JMS**. You can indicate the fields that will be available for each of the built-in types. Mark required fields as mandatory and indicate whether to display them in the **Details** or **Interaction** tab.

Click **New** to create a new component type. Specify a **Name**, **Assigned Icon**, **Supported file extensions**, and **Contract Parser ID**. Select the fields to display for the new component, for example in its **Details** tab. You can also mark fields as mandatory and indicate whether they will be visible in the **Interaction** tab.

For user interface details, see the "Component Types Tab" on page 181.

Customize group permissions - optional

Assign the permissions for the component-related tasks per user group. For a list of the component-related tasks that you can configure, see "Groups and Permissions" on page 187.

  a Click **Groups and Permissions** in the left pane**.**

  b Click **New Group** to open the New Group dialog box. Specify a name and permission level. Click **OK**.

  c Select the **Permissions** tab and click the **Application Components** subtab. The subtabs are in alphabetical order.

  d Modify the permission levels as required.

  e Click **Save**.

Customize application component module access - optional

Indicate which user groups will have access to the **Application Components** module.

    a   Click **Module Access** in the left pane.

    b   Expand the window or scroll to the right to view the **Application Components** column.

    c   Select the check boxes as required. To enable or disable permissions for an entire column, select or clear the check box in the header row.

    d   Click **Save**.

Customize project entities - optional

Use the Project Entities section to define new fields for the application component, changes, and groups.

    a   Click **Project Entities** in the left pane.

    b   Expand the node to which you want to add or modify an entity. **Application Component**, **Application Component Change**, or **Application Component Group**.

    c   To modify a built-in field, expand the **System Fields** node and select a field. You can modify the label and indicate if the field should be required.

    d   To create a new field, select the **User Fields** node and click the **New Field** button. Provide a label, type, and maximum string length for **String** type fields. Select any other relevant options as described in the "Project Entities Pane" on page 184.

    e   To create a new Memo type field, expand the **New Field** button and select **New Memo Field.**

    f   Click **Save**.

Reference
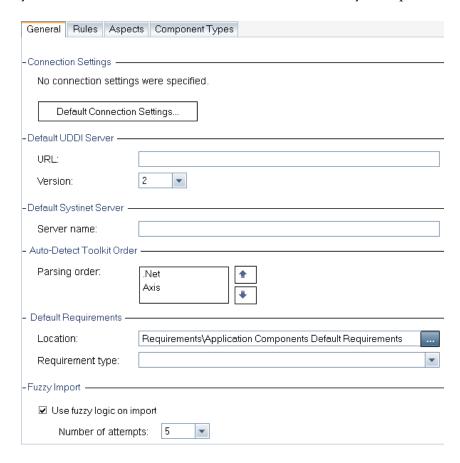
🔍 Project Customization User Interface

This section includes (in alphabetical order):

➤ General Tab on page 172

➤ Rules Tab on page 175

➤ Aspects Tab on page 179

➤ Component Types Tab on page 181

➤ Project Entities Pane on page 184

➤ Groups and Permissions on page 187

➤ Module Access Pane on page 191

🔍 General Tab

This tab enables you to customize the settings for parsing contracts and importing WSDLs. You can enter a default server and credentials, so that you will not need to enter the information each time you import a service.

| General | Rules | Aspects | Component Types |

Connection Settings

No connection settings were specified.

[Default Connection Settings...]

Default UDDI Server

URL: [   ]

Version: [2   ▼]

Default Systinet Server

Server name: [   ]

Auto-Detect Toolkit Order

Parsing order:
| .Net |
| Axis |
⬆
⬇

Default Requirements

Location: [Requirements\Application Components Default Requirements] [...]

Requirement type: [▼]

Fuzzy Import

☑ Use fuzzy logic on import

Number of attempts: [5   ▼]

To access	Do the following:
	1 Select **Tools** > **Customize**.
	2 Select **Application Components** in the left pane.
	3 Select the **General** tab.
Important information	To resume work in the ALM modules, click the [Return] button in the top right corner.
Relevant tasks	"Customize the General settings - optional" on page 168

User interface elements are described below:

General Tab Sections	Description
Connection Settings	➤ **Default Connection Settings**. Opens the Connection Setting dialog box for setting the default connection settings. For details, see "Connection Settings Dialog Box" on page 76.
Default UDDI Server	➤ **URL.** The default address of the UDDI server when a user opens the Select Service from UDDI Dialog Box, for example http:/<*my_server*>.<*my_domain*>:8090/juddi/inquiry. ➤ **Version.** The UDDI version: **2** or **3**.
Default Systinet Server	**Server Name.** The default path of the Systinet server when a user opens the Select Service from Systinet Dialog Box.
Auto-Detect Toolkit Order	**Parsing Order.** The order in which to parse a WSDL in auto-detection, using the .Net or Axis toolkits.

General Tab Sections	Description
Default Requirements	➤ **Location.** The default requirement to use when linking a component to a test. The **Browse** button opens the Select Requirement dialog box.
	➤ **Requirement Type.** One of the standard requirement types: **Undefined, Functional**, **Testing**, **Performance**, or **Business Model**.
Fuzzy Import	➤ **Use fuzzy logic on import.** For .NET services, use a looser set of criteria for parsing the contract.
	➤ **Number of Attempts.** The number of attempts to try parsing the contract before aborting the import. The default is 5.

Select Requirement

This dialog box lets you select a default requirement.
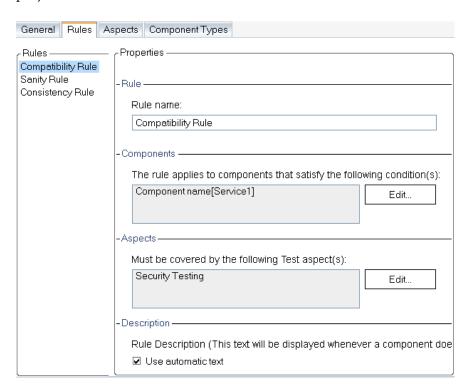
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements (A-Z)	Description
	Refresh All. Refreshes all of the requirements in the tree.
	Set/Clear Filter. Opens a drop down list with the following options: ➤ **Set Filter / Sort.** Opens the Set Filter/Sort dialog box allowing you to filter and sort the requirements. ➤ **Clear Filter / Sort.** Clears all filters and sorting settings and displays all requirements.
	New Requirement. Opens the New Requirement dialog box for defining a new requirement.

UI Elements (A-Z)	Description
📁	**New Requirement Folder.** Creates a new requirement folder, under the selected folder. **Note:** This is only available when selecting a folder—not a requirement.
<Requirements Tree>	A tree hierarchy of all the requirements.

🔖 Rules Tab

This tab lets you define rules for the application components in your project.

To access	Do the following: **1** Select **Tools** > **Customize**. **2** Select **Application Components** in the left pane. **3** Select the **Rules** tab.
Important information	To resume work in the ALM modules, click the ![Return] button in the top right corner.
Relevant tasks	"Customize the component rules - optional" on page 168
See also	"Alerts Dialog Box" on page 33

User interface elements are described below:

UI Elements	Description
Toolbar buttons	➤ **Save.** Saves the changes to the **Rules** tab. ➤ **New.** Opens the Create New Rule dialog box. ➤ **Delete.** Removes the selected rule.
Properties	The properties of the rule: ➤ **Rule.** The rule name. If a component violates a rule, the Alert box lists it by this name. ➤ **Components.** The rule is only applied to components that meet the selected conditions. To set a condition, click the **Edit** button to open the Filter Application Components Dialog Box described on page 177. To create a global rule that will apply to all components, specify a filter condition that is common to all components, for example, Component ID > 0. ➤ **Aspects.** The testing aspects that must be covered for the component. To select the aspects, click the **Edit** button to open the Select Aspects dialog box described on page 163. **Note:** You can only select a direct descendent of the root aspect. ➤ **Description**. Text describing the rule. To allow Service Test Management to generate a description based on the selected aspects and conditions, select **Use automatic text**. If the rule is violated, the Alerts dialog box shows this text.

Filter Application Components Dialog Box

User interface elements are described below:

UI Elements	Description
	Clear Filter. Clears all conditions to which the rule should apply.
	Copy filter. Copies the current filter settings onto the clipboard. You can paste this filter for another rule or save it to a text file for later use.
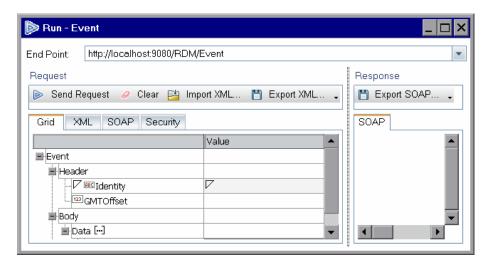
UI Elements	Description
📋	**Paste filter.** Uses a filter from the clipboard.
Filter tab	A list of field names and their conditions. The field names correspond to the component fields in the Project Entities list. **Tip:** Click the down arrow in the right side of the row, to open the Select Filter Condition dialog box. You can manually type the condition that must be met and use asterisks (*****) as wildcards.
Cross Filter tab	Allows you to filter the following entities within the Application Components module: **Groups**, **Requirements**, **Tests**, **Defects**, **Application Changes**, and **Alerts**. For Alerts: Enable the **Show application components with alerts** option to hide all components without alerts.

Aspects Tab

This tab lets you define the properties of the built-in aspects and define new ones for the project.

This dialog box enables you to run a spot test for an individual operation. You can set up a request and view the SOAP response to verify if an individual operation is functional.

To access	Do the following:
	1 Select **Tools** > **Customize**.
	2 Select **Application Components** in the left pane.
	3 Select the **Aspects** tab.
Important information	To resume work in the ALM modules, click the `Return` button in the top right corner.
Relevant tasks	"Customize the aspects - optional" on page 168
See also	"Alerts Dialog Box" on page 33
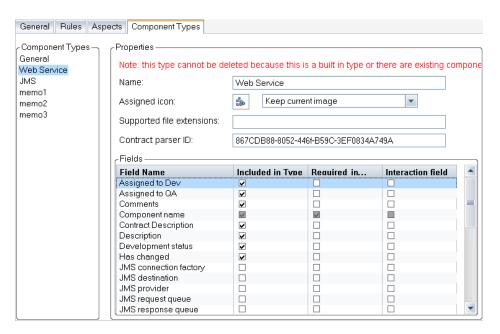
User interface elements are described below:

UI Elements	Description
🖫 Save	Saves the Rule settings.
New	Opens the Create New Aspects dialog box.
Delete	Removes the selected aspect from the list. You can only delete user-defined aspects—not built-in ones.
Aspects	A list of the aspects that are available for the current project.
Properties	The properties of the aspect: ➤ **Aspect Name.** The aspect name as it will appear in the tree. For predefined system aspects, this field is read-only. ➤ **Aspect Description.** Text describing the goal of the aspect. ➤ **Requirement type.** The requirement type:—a drop-down of predefined requirements, such as **Business**, **Group**, or **Performance,** and any user-defined requirements. For details, see the *ALM User Guide*. ➤ **Requirement name.** A name for the requirement. To include the component name, use the variable %service_name%. The **Requirement and Test Generation** wizard will use this name when creating a requirement for this aspect. ➤ **Requirement description**. Text describing the requirement. To include the component name, use the variable %service_name%.

🔍 Component Types Tab

This tab provides information about the different application component types. You can control which fields will be available in the **Details** and **Interaction** tabs for each component type and if they are mandatory. You can also define new component types.

To access	Do the following:
	1 Select **Tools > Customize**.
	2 Select **Application Components** in the left pane.
	3 Select the **Component Types** tab.
Relevant tasks	"Manage the component types - optional" on page 169

UI Elements (A-Z)	Description
New	Adds a new component type to the list. To rename the component, enter text in the right pane's **Name** field.
Delete	Removes the selected non- built-in component from the list. **Note**: You cannot remove a type if a component of that type exists in the repository.
Component types	A list of the built-in and custom component types. The built-in types are **General**, **Web Service**, and **JMS**.

UI Elements (A-Z)	Description
Properties	The component type details: ➤ **Name**. The name of the component type as it will appear in ALM. ➤ **Assigned icon**. The icon representing the component of this type in the application components tree. A drop-down includes several options. For details, see "Application Components Module Icons" on page 55. ➤ **Support file extensions**. The file extensions supported as a contract for components of this type, for example, WSDL. ➤ **Contract parser ID**. The COM object of the parser GUID used for parsing a component definition file, such as the WSDL. If a a parser has this unique ID and is registered on the client machine, Service Test Management uses it to parse contract files of this type.
Fields	The fields that will be available in the component's Details and Interaction tabs. These settings are specific for each component type—both built-in and user-defined. ➤ **Included in Type**. Show the field in the components's Details tab, but do not make it mandatory. ➤ **Required for Type**. Show the field in the components's Details tab, and make it mandatory. ➤ **Interaction Field.** Show the field in the **Interaction** section of the Add New Component dialog box and **Interaction** tab. For details, see the "New Application Component Dialog Box" on page 51.
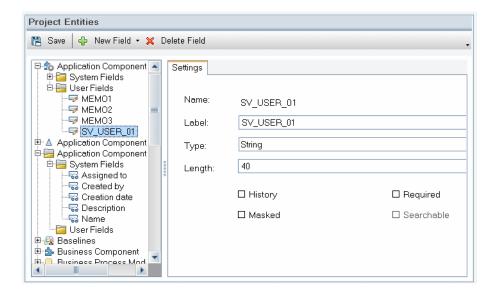
🔦 Project Entities Pane

This pane lists the project entities. Each entity contains system fields and user-defined fields:

➤ **System fields.** These are ALM default fields. You cannot add or delete system fields—you can only modify them.

➤ **User fields.** Customizable user-defined fields. You can add, modify, and delete user-defined fields.

For detailed information on ALM entities and fields, refer to the *HP ALM Database Reference*.

This dialog box enables you to run a spot test for an individual operation. You can set up a request and view the SOAP response to verify if an individual operation is functional.

To access	Do the following:
	1 Select **Tools** > **Customize**.
	2 Select **Project Entities** in the left pane.
	3 Expand the **Application Component**, **Application Component Group**, or **Application Component Change** entities.
Important information	You can only create and delete **User** fields, but not **System** fields. For System fields, you can modify the label and select some additional options, depending on the field.
Relevant tasks	"Customize project entities - optional" on page 170
See also	"Application Component Details Pane" on page 46

User interface elements are described below and in greater detail in the *HP ALM User Guide*.

UI Elements	Description
➕ New Field ▾	Opens a drop-down list allowing you select **New Field** or a **New Memo Field.**
✖ Delete Field	Removes the selected user-defined field.
Name	A read-only field name used in the ALM database table.
Label	The name of the field to display in the Project Entities tree and the component's **Details** tab. This label is referenced anywhere the field is used, such as in the filter criteria, error messages, and graphs.
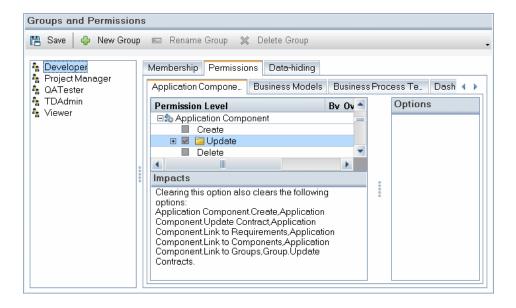
UI Elements	Description
Type	The type of data valid for the field: ➤ **Number.** Enables integer entry only. ➤ **String.** Enables the entry of any character string. ➤ **Lookup List.** Displays the Lookup List area and enables the selection from a drop-down list. ➤ **User List.** Enables the selection of a user name from your ALM users list. ➤ **Date.** Enables the selection of a date. ➤ **Memo.** Enables the entry of blocks of data. Note that you can add up to 5 memo fields to each ALM entity (available by selecting **New Field** > **New Memo Field** from the toolbar).
Length	The field size (available for **String** type only). **Note:** The maximum field length is 255 characters.
Goto List	Displays a predefined list. (Available only when the **Lookup List** type is selected.) To open a predefined list, select a list from the Lookup List box and click the **Goto List** button.
History	Preserves a log of values entered in the selected field.
Masked	The input data mask for the field (available only when the **String** type is selected).
Required	Indicates that a user must enter a value for the field.
Searchable	Indicates a searchable field (available only when the **Text Search** option is enabled in the **DB Servers** tab).
Lookup List	A list of predefined lists (available only when the **Lookup List** type is selected). To associate a field with a predefined list, select a list from the Lookup List box. To view or modify the selected list, click the **Goto List** button.

UI Elements	Description
Verify value	Limits the user to select a value only from the items that are listed in the list box (available when **Lookup List** or **User List** is selected).
Allow multiple values	Allows the user to select more than one value in any field that is associated with a predefined lookup list (available only when selecting a **Lookup List** type field).

🔍 Groups and Permissions

This customization section lets you define new user groups and set their permissions.

To access	Do the following:
	1 Select **Tools** > **Customize**.
	2 Click **Groups and Permissions** in the left pane.
	3 Select a group and click the **Permissions** tab.
	4 Select the **Application Components** tab (the subtab names are in alphabetical order).
Important information	To resume work in the ALM modules, click the [Return] button in the top right corner.
Relevant tasks	"Customize group permissions - optional" on page 169

The **Application Components** tab displays the following permissions for elements available in the **Application Components** module:

Element / Entity	Permission Level
New Group	Opens the New Group dialog box for creating a new permission group.
Rename Group	Allows you to rename the selected user-defined group.
Delete Group	Removes the selected user-defined group.
ⓘ	**Alert.** This icon indicates that selecting or clearing the current permission, will impact other permissions. The mouseover message provides the impact details. ☑ Create ⓘ
Permission Levels - Application Change	Lets you set the group's permissions for handling changes in an application component. You can enable or disable the following capabilities: ➤ **Create.** Create a new change. ➤ **Update.** Update fields in an existing change. For details, expand the node and see "Application Change Details Dialog Box" on page 207 ➤ **Delete.** Delete a change.

Element / Entity	Permission Level
Permission Levels - Application Component	Lets you set the group's permissions for creating and modifying application component details. You can enable or disable the following capabilities: ➤ **Create.** Create a new change. ➤ **Update.** Update fields in an existing component. For details, expand the node and see "New Application Component Dialog Box" on page 51. ➤ **Delete.** Delete a component. ➤ **Link to Components.** Link to another application component. ➤ **Link to Groups**. Link a component to an Application Component group. ➤ **Link to Requirements**. Link a component to existing requirements. ➤ **Update Contract**. Update a component's contract from its source.
Permission Levels - Folder	Lets you set the group's permissions for creating and removing folders in the application components tree. You can enable or disable the following capabilities: ➤ **Create.** Create a new folder. ➤ **Update.** Update fields for an existing folder. The available fields are: **Attachment**, **Description**, **Father ID**, **Folder ID**, **Logical Path**, **Name**, and **UDF**. ➤ **Delete.** Delete a folder.

Element / Entity	Permission Level
Permission Levels - Group	Lets you set the group's permissions for creating and modifying Application Component groups. You can enable or disable the following capabilities. ➤ **Create.** Create a new group. ➤ **Update.** Update fields in an existing group. For details, expand the node and see "New Group Dialog Box" on page 103. ➤ **Delete.** Delete a group. ➤ **Link to Requirements**. Link a group to existing requirements. ➤ **Update Contracts**. Update the contracts in a group's components from their source.
Permission Levels - Customize	Provides the members of the group access to the **Application Components** section in Customization.
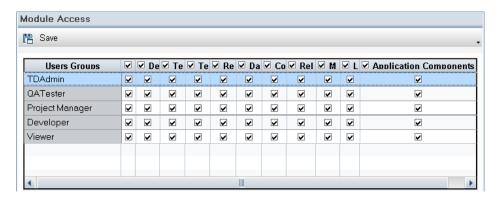
Module Access Pane

This pane lists the module access per group. You can control access for the various ALM modules.

For detailed information on module access, refer to the *HP ALM Database*

To access	Select **Tools** > **Customize** and click **Module Access** in the left pane.
Relevant tasks	"Customize application component module access - optional" on page 170
See also	"Customize group permissions - optional" on page 169

The Module Access pane contains a table indicating each group's permission per module.

Selecting the check box in the header row, checks all rows in that column. Clearing the check box in the header row, clears all rows in that column.

For more details about Module Access permissions, see the *HP Application Lifecycle Management User Guide*.

10

Change Analysis

This chapter includes:

Concepts

Tasks

Reference

Concepts

⚙ Change Analysis Overview

Change impact analysis helps you assess the impact of changes to the application components. For each change, Service Test Management displays a description of the change, an assessment of the change-risk, and the change status.

You can introduce changes manually and specify the change details. If you know of changes in the component's contract, you manually insert a new change and provide details about the change and its risk.

Service Test Management also generate a change list automatically. When you update an application component or a application group, it compares the latest definition with the definition stored in ALM. If there are changes at the operation signature level or in the data structure of an argument, Service Test Management lists them in the **Changes** tab.

In addition to the above methods, developers can register changes through a IDE plug-in.

! Service Test Management flags changed application components, groups, or folders, in the application components tree with an **Alert** icon.

Tip: A green asterisk on the **Changes** tab indicates a new change.

To verify the changes, you can generate a test set. These test sets can contain all tests linked to the updated component, or only tests that use the operations affected by the changes.

Each test set must be linked to a requirement. If a component does not have requirement coverage, Service Test Management lets you link it to a default requirement.

After creating or generating change-based requirements and a test set, you can run the linked or affected tests in the test set to verify whether these changes caused regression in functionality or performance. You can then determine which entities were effected and who is responsible for them.

To determine if a change violates a rule associated with a component, see the "Rules Tab" on page 133.

🔵 Reporting Changes Manually

Service Test Management allows you to define changes manually. This change can indicate an update in the contract file or other changes that affected the contract file.

You can also add descriptions and comments, and assign the change to developers or QA personnel. For details, see "New Application Change Dialog Box" on page 204.

🔵 Reporting Changes by Updating Components

You can use the **Update Component** feature to check for changes in a component. If changes are found, the component's definition stored in ALM is updated. You can update the definition from the original location or from a custom location.

For Web Service components, you must use the original location type: File, URL, UDDI, or Systinet. You can update WSDLs that reside on secure locations and WSDLs accessed through proxy servers using the authentication credentials. For details, see the "Secure Services and Proxy Servers" on page 65.

Note: When updating a Web service from a UDDI or Systinet server, it must have the same name as the one in the original WSDL used to import the service.

You can update individual components through the **Components** menu. When you update a component, the **Changes** tab lists the changes. Service Test Management flags the application component with an Alert icon, in the Component tree. For details, see "Review the changes" on page 199.

Note: If **Keep up to date** is set to Yes in the **Details** tab, the component definition is automatically updated from the original source each time it is loaded.

Tasks

🔧 How to Perform Change Analysis

This task describes how to create and review the changes for your application component.

This task includes the following steps:

➤ "Define a New Change - optional" on page 197

➤ "Update a Component - optional" on page 198

➤ "Review the changes" on page 199

➤ "Evaluate the tests" on page 199

➤ "Create regression tests" on page 199

➤ "Run the test sets" on page 200

➤ "Check the test status" on page 200

➤ "Update the change status" on page 200

1 Define a New Change - optional

To manually define a change in the Changes list:

    a In the application components tree, select a component.

    b Select the **Changes** tab and click the New Change button.

    c In the New Application Change dialog box, make sure that the following mandatory fields have values:

        ➤ **Subject**. A meaningful name for the change.

        ➤ **Affected part**. The name of the affected application component part.

        ➤ **Affected part type.** The type of part affected by the change: Component, Field, Interface, or Operation.

➤ **Change Status.** The change's current status: In Test, New, Obsolete, or Modified.

➤ **Change type.** The nature of the change: Added, Updated, or Removed.

➤ **Risk.** The risk of the change: High, Medium, or Low.

d Enter values into the optional fields, such as **Description**, **Assigned to**, and so forth.

2 Update a Component - optional

To automatically generate a list of changes, update the component contract stored in ALM with the latest contract file. Service Test Management flags the changed component in the application components tree.

a In the application components tree, select the component you want to update.

b Choose one of the following update options:

➤ To update a component using the same contract location, choose **Components** > **Update Component** > **Update Component**.

➤ To update a component using a different location, choose **Component** > **Update Component** > **Update Component from**. Browse for a newer version of the contract in another location.

The update process begins. Service Test Management checks for changes in the contract and informs you if the component was updated.

3 Review the changes

Select an application component that has been updated, indicated by an alert icon ❗. Click the **Alert** icon to open the Alerts dialog box and view a list of the alerts. Click **Show** to open the selected change in the **Changes** tab. The grid lists the changes that were made to the component. For details, see the "Application Change Details Dialog Box" on page 207.

4 Evaluate the tests

Evaluate the tests that are associated with the component in the **Testing Status** tab, and add additional tests if necessary. For more information about tests linked to components, see "View the Tests coverage - optional" on page 129.

5 Create regression tests

Click **Create Test Set** to open the wizard that creates tests to analyze the impact of change. For details, see the "Create Change Impact Test Set Wizard" on page 210.

Tip: The **Create Change Impact Test Set** wizard does not allow you to enter values for required Test Set fields. To enter values for these fields, open the **Test Lab** module, go to the test set, and enter values for the required fields.

---

6 Run the test sets

Run the test sets from the **Test Lab** module. You can verify the component behavior and analyze the test results using ALM graphs, reports, and documents. For details on running test sets and analyzing test results, refer to the *HP Application Lifecycle Management User Guide*.

7 Check the test status

Review the coverage status of the parent requirement created for the changes set in the **Testing Status** tab, to verify that the coverage includes the latest changes. For details, see ""Requirements Coverage" on page 122.

8 Update the change status

Verify the behavior of the changed component and manually update the change status in the **Changes** tab through the following steps:

a Select an updated component in the application components tree, and click the **Changes** tab.

b In the changes list, select the change whose status you want to update.

c Double-click on the change to open the Application Change Details dialog box. In the **Change Status** box, select a change status:

➤ **Verified.** if the change was tested and did not cause a regression.

➤ **Obsolete.** if the change does not affect the component.

For details about change status, see the "Application Change Details Dialog Box" on page 207.

Reference
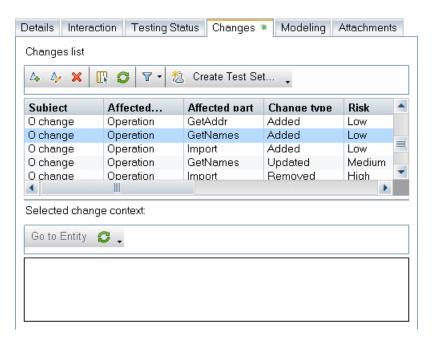
🔍 Changes User Interface

This section includes:

➤ Changes Tab on page 202

➤ New Application Change Dialog Box on page 204

➤ Application Change Details Dialog Box on page 207

➤ Create Change Impact Test Set Wizard on page 210

➤ Tests Tree Panel on page 213

🔍 Changes Tab

This tab allows you to view and manage the changes made to the application component.
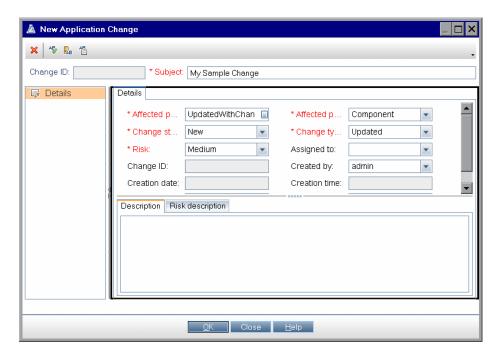
To access	Do the following:
	1 Select the Application Components module. **2** Select an entity in the application components tree. **3** Click the **Changes** tab.
Relevant tasks	"Review the changes" on page 199
Important Information	For folders and groups, the **Changes** tab is read-only.

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements	Description
	New Change. Opens the New Application Change dialog box to create a new change upon your component. See the "New Application Change Dialog Box" on page 204.
	Change Details. Opens the Application Change Details for viewing and editing. See the "Application Change Details Dialog Box" on page 207
	Delete. Removes a change from the Changes grid.
	Select Columns. Enables you to select the columns to display in the Changes grid. For a description of the columns, see the "Application Change Details Dialog Box" on page 207.
	Refresh. Reloads the list of changes.
	Set Filter/Sort. Enables you to set a filter for the changes or sort the changes by a criteria.
🧪 Create Test Set...	**Create Test Set.** Opens a wizard for creating a test set. For details, see "Create Change Impact Test Set Wizard" on page 210.
<Changes grid>	A list of all the changes for the selected component to which the filter applies. To clear the filter, use the Set/Filter sort button. To arrange the columns, use the Select Columns button. **Default:** Subject, Affected part type, Affected part, Change type, Risk, Change status, Creation date, Assigned to, Release, and Cycle.

🔍 New Application Change Dialog Box

This dialog box lets you define a new change for your application component.

To access	Select the **Changes** tab and click the **New Change** ⟁ button.
Relevant tasks	"Review the changes" on page 199
See also	"Changes Tab" on page 202

User interface elements are described below:
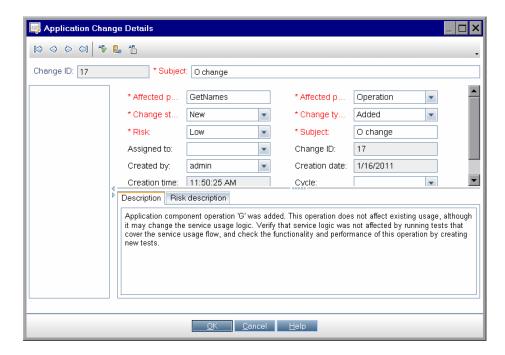
UI Elements (A-Z)	Description
	Clear all Fields. Clears all the fields in the dialog box.
	Editing buttons: **Spell Check**, **Thesaurus**, and **Spelling Options**. For details, see the *ALM User Guide*.
Affected part	The name of the part affected by the change, such as the component name or operation name.
Affected part type	The part type affected by the change: **Component**, **Field, Interface**, or **Operation**.
Assigned to	The user name of the person who is responsible for the change.
Change ID	A unique identification string for the change. This field is read-only and is generated when you save a new change. This is visible in the Application Change Details dialog box.
Change status	The current status of the change. By default, the change status is **New**. Update the status throughout the change analysis process as follows: ➤ **In test.** The change is being tested. ➤ **New.** There is a change in the component. ➤ **Obsolete.** The change does not affect the component. ➤ **Verified.** The change was tested and did not cause a regression.
Change type	The type of change is indicated as follows: ➤ **Added.** New functionality was added. ➤ **Removed.** Functionality was removed. ➤ **Updated.** Changes were made to existing functionality. Details of these changes are displayed in the Description column.
Created by	The user name of the person who created the changes.

UI Elements (A-Z)	Description
Creation date	The date on which the change was made. By default, the creation date is set to the current database server date. This field is read-only and is generated when you save a new change.
Creation time	The time at which the change was made. This field is read-only and is generated when you save a new change. **Default:** The current database server time.
Cycle	The testing cycle in which the change was introduced or in which the change needs to be verified.
Description	A description of the change and suggested tests to verify them. For operation changes (where **Part Type** is **Operation** and **Change Type** is **Updated**), the change description contains details of the argument that was added or removed from the operation.
Release	The planned release version.
Risk	The business risks and costs associated with the change. A change can have one of the following risk levels: ➤ **High.** Changes that are likely to affect functionality and fail any test that is not updated. High-risk changes include operation removal and parameter changes. ➤ **Medium.** Changes that may affect functionality, but may not necessarily fail existing tests. Medium-risk changes include component changes (notification that some of the component files have changed). ➤ **Low.** Changes that are unlikely to fail or affect existing functionality or implemented tests. Low-risk changes include operation additions (as they do not affect existing functionality), service address changes (deployment only), and port type changes.

UI Elements (A-Z)	Description
Risk Description	A textual description of the risk. Tip: Click **Add Comment** to create a remark with a name, time, and date stamp.
Subject	A short title describing the change. This field is mandatory.

🔖 Application Change Details Dialog Box

This dialog box enables you to view and manage the changes applied to the component.

To access	Select the **Changes** tab and click the **Change Details** 🖉 button.
Relevant tasks	"Review the changes" on page 199
See also	"Changes Tab" on page 202.

User interface elements are described below:

UI Elements(A-Z)	Description
◁◁ ◁ ▷ ▷▷	**Change Navigation**. Enables navigation between the changes for display in the Application Change Details dialog box. The buttons perform the following actions, from left to right: ➤ **Go to first change** ➤ **Go to previous change** ➤ **Go to next change** ➤ **Go to last change**
ᴬᴮ 📖 ᴬᴿ	Editing buttons: **Spell Check**, **Thesaurus**, and **Spelling Options**. For details, see the *ALM User Guide*.
Affected part	The component part affected by the change.
Affected part type	The part affected by the change: **Component, Field, Interface**, or **Operation**.
Assigned to	The user name of the person responsible for the change.

UI Elements(A-Z)	Description
Change status	The current status of the change. By default, the change status is **New**. Update the status throughout the change analysis process as follows: ➤ **In test.** The change is being tested. ➤ **New.** There is a change in the component. ➤ **Obsolete.** The change does not affect the component. ➤ **Verified.** The change was tested and did not cause a regression. **Tip:** To eliminate the Alert icon next to the component, change the status of all changes to **Verified** or **Obsolete**.
Change type	The type of change is indicated as follows: ➤ **Added.** New functionality was added. ➤ **Removed.** Functionality was removed. ➤ **Updated.** Changes were made to existing functionality. Details of these changes are displayed in the Description column.
Created by	The user name of the person who ran the update that created the change.
Creation date	The date on which the change was made. By default, the creation date is set to the current database server date.
Creation time	The time at which the change was made. **Default:** the database server time at the time of the change.
Cycle	The testing cycle in which the change was introduced or in which the change needs to be verified.
Description	A description of the change and suggested tests to verify. For operation changes (where **Affected Part Type** is **Operation** and **Change Type** is **Updated**), the change description contains details of the argument that was added or removed from the operation.
Release	The planned release version.

209

UI Elements(A-Z)	Description
Risk	The business risks and costs associated with the change. A change can have one of the following risk levels: ➤ **High.** Changes that are likely to affect functionality and fail any test that is not updated. High-risk changes include operation removal and parameter changes. ➤ **Medium.** Changes that may affect functionality, but may not necessarily fail existing tests. Medium-risk changes include changes (notification that some of the component files have changed). ➤ **Low.** Changes that are unlikely to fail or affect existing functionality or implemented tests. Low-risk changes include operation additions (as they do not affect existing functionality), service address changes (deployment only), and port type changes.
Risk Description	A textual description of the risk. Tip: Click **Add Comment** to create a remark with a name, time, and date stamp.
Subject	A short title describing the change.

🔍 Create Change Impact Test Set Wizard

This wizard enables you to create a test set for determining the impact of changes made to an application component.

To access	Select > **Changes tab** > 🧪 Create Test Set... ▾ .
Relevant tasks	"Create regression tests" on page 199
Wizard map	The Create Change Impact Test Set wizard contains: Coverage Page > Test Set Page

🔖 Coverage Page

This page allows to set the scope of the test set.

Wizard map	The **Create Change Impact Test Set** wizard contains: **Coverage Page** > Test Set Page

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements (A-Z)	Description
	Select Columns. Enables you to select the columns to display in the **Changes** list.
Select All	Selects all records in the **Changes** list.
Component Coverage	The components to cover with the test set. Either of the following options: ➤ **<component_name> only** ➤ **<component_name> and all dependent components**
Scope	The scope of the test set: ➤ **All tests.** Create a test set using all tests linked to the component. ➤ **Test affected by the following changes.** Create a test set only using the test that was affected by the selected change(s). Select the changes you want to analyze. Make a multiple selection using the **Ctrl** key.
<Changes list>	A list of all the changes for the selected component. To set the columns to display, use the Select Columns button. **Default:** Subject, Affected Part Type, Affected Part, Change Type, Risk, Change Status. Creation Date, Assigned To, Release, and Cycle.

🔍 **Test Set Page**

This wizard helps you create a test set to determine the impact of your changes to the component. The test set will contain all the tests you need to run, in order to establish that your change did not affect the functionality of the component.
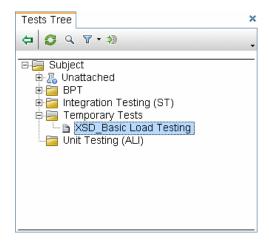
Wizard map	The **Create Change Impact Test Set** wizard contains:
	Coverage Page > **Test Set Page**

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements (A-Z)	Description
🔼 Select	**Select.** Opens the Test Plan tree to enable you to select tests for the test set. For details, see the "Tests Tree Panel" on page 213.
✖	**Delete.** Removes the selected test from the Tests list.
...	**Browse.** Opens the Select Test Set folder dialog box.
	Tip: In the Select Test Set folder dialog box, click the **New Folder** button 📂 to create a new folder.
Folder	The folder name for the test set. Use the **Browse** button to locate a folder.
Name	A name for the test set.

🔍 Tests Tree Panel

This tab enables you to add tests to a test set.

To access	Do the following:
	1 Select a component and click the **Changes** tab.
	2 Select a change in the list and click the ⚗ Create Test Set... ▾ button.
	3 In the Create Change Impact Test Set wizard, specify the coverage details.
	4 Click **Next** to advance to the Test Set wizard page.
	5 Click the 🔲 Select button.
Relevant tasks	"How to Perform Change Analysis" on page 197
See also	"How to Link Components to Requirements and Tests" on page 149

The following elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
⬅	**Add to Coverage.** Adds the selected test to the test list.
🔄	**Refresh.** Refreshes the Tests tree.
🔍	**Find.** Opens the Find dialog box for locating a test by its name or another criteria. For details, click F1 in the Find dialog box.
▼ ▾	**Set/Clear Filter.** Opens a drop-down list with the following options: ➤ **Set Filter / Sort.** Opens the Set Filter/Sort dialog box allowing you to filter and sort the tests. ➤ **Clear Filter / Sort.** Clears all filters and sorting settings and displays all tests.
⤳	**Go to Test by ID**. Locates the test whose ID you specify.
✕	Closes the **Tests Tree** panel. To reopen this view, click the **Select** button.
\<Tests tree\>	A tree view of all of the tests. **Tip:** If a filter is active, there could be hidden components. To see all components, click **Clear Filter**.

11

Reports and Graphs

This chapter includes:

Concepts

➤ Service Test Management Reports and Graphs Overview on page 216

Tasks

➤ How to Create and Generate Reports on page 225

➤ How to Generate Graphs on page 227

Concepts

🍀 Service Test Management Reports and Graphs Overview

ALM's **Dashboard** module provides you with several ways to generate summary reports.

Using the standard Document Generator, you can create summary reports of your test plan and requirements. Use the **Tools** menu to open the **Document Generator**. For details, see the *ALM User Guide*.

This section describes the Excel reports that you can generate through ALM, that are specific to Service Test Management.

Using the capabilities of Excel, you can create data tables and display the results in graphical form. You can also create custom reports, by designing Excel-compatible SQL queries.

You can use the data you extract from a query for external applications, such as running a Visual Basic script to process and analyze the data.

This section also includes:

➤ "Built-in Reports" on page 217

➤ "Application Component Query Builder Entities" on page 218

➤ "Application Component Graphs" on page 219

➤ "Application Changes Graphs" on page 222

Built-in Reports

Service Test Management provides the following pre-defined reports for application components:
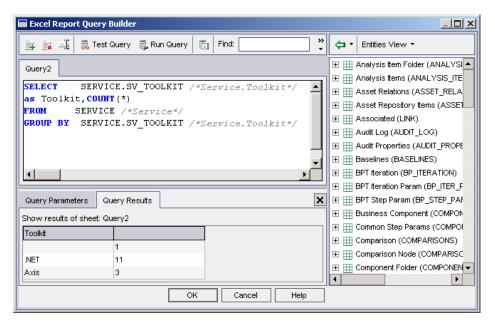
Report Name	Description
Active Changes	Exports the project's application components and their list of active changes.
Application Component Defects	Exports the project's application components and the list of new or open defects which are linked to them.
Application Component Tests	Exports the project's application components and the list of tests which either use or cover them.
Application Components Per Responsible User	Exports the count of application components per a responsible user and displays this data as an Excel chart
Coverage	Exports the requirement coverage of each application component and the total coverage. The results are displayed in a table and in a pie chart.
Outstanding Changes	Exports the list of changes in the application components which are in the 'New' or 'In Test' status.
Rule Violations	Exports the list of all rules together with components which violated them.
Web Service Components by Toolkit	Exports the distribution of toolkits which are used by the Web services.
Web Service Components by Type	Exports the distribution of Web service import types: File, URL, UDDI, or Systinet.

You can generate the reports as is, or modify them as necessary in the Query Builder. For details, see "Create a custom report based on a pre-defined report" on page 226.

Application Component Query Builder Entities

You can create custom queries on the project's database by choosing application component related entities from the Entities list. You can add these queries to an existing report, or create a new one.

Service Test Management adds the following entities to Query Builder's Entities View: **Component Change to Defects**, **Component Change to Requirements**, **Component Folder**, **Component Param**, **Component Step**, **Component Type**, **Component Type to Field Relation**, **Service Aspect**, **Service Dependency**, **Service Folder**, **Service Group to Requirement Relation**, **Service Operation to Component Relation**, **Service Operation to Test Relation**, **Service to Group Relation**, **Service to Requirement Relation**, **WSDL Operation**, **WSDL Operation Dependency**, and **WSDL Operation Parameters**. You can create both private queries or public ones that you can share with other users.

For example, you could create a pivot table to summarize component-related data, such as to display in tabular format the status count for the different coverage statuses.

Once you generate an Excel report, you can use the full capabilities of Excel to display the results in graphical form, such as bar or pie charts.

For task details, see "How to Generate Graphs" on page 227.

Application Component Graphs

You can create custom graphs for the Application Component module using Service Test Management's built-in graphs. You can create **Progress**, **Summary**, and **Trend** graphs for the application components and for the application component changes.

The graph's **Configuration** tab lets you customize the graphs by setting the x and y axis, filtering the data, and grouping the results.

Summary Graphs
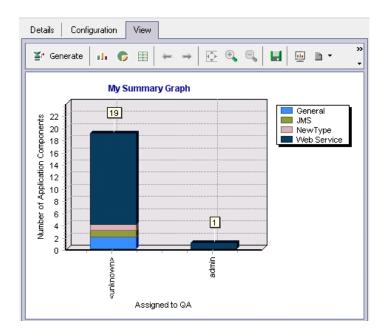
In Application Component Summary graphs, you can configure the **x-axis** to display one the following fields: Assigned to Dev, Assigned to QA, Authentication default credentials, Created by, Development status, Has changed, Import type, JMS provider, JMS transport, JNDI initial context factory, Keep up to date, Messaging model, Override address, Proxy default credentials, Toolkit, Use authentication, Use proxy, Visibility, QC Project, or other fields that you defined in customization.

You can configure the **y-axis** to display

➤ A **Count** of the Application components

➤ A **Sum of** the custom fields, defined in the Customization module

Service Test Management also allows you to filter the graph by all system and UDF fields defined for the Application Component entity.

The following **Summary** graph shows the number of components assigned to QA as a function of the total number of components.

Progress Graphs

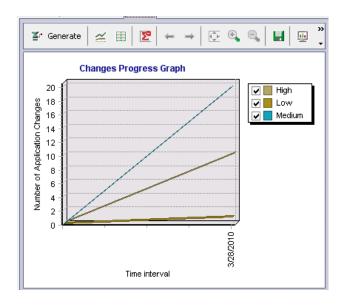You can set up a **Progress** graph to plot the number of application components against a time interval.
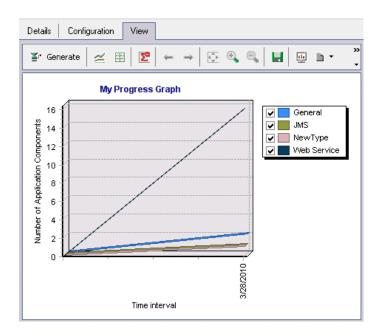
You can configure the **x-axis** to display a time interval: Last 7 days, Last 5 weeks, Last 12 months, or All days.

You can configure the **y-axis** to display

➤ A **Count** of the Application components

➤ A **Sum of** the custom fields, defined in the Customization module

You can group the results by the following criteria: Assigned to Dev, Assigned to QA, Authentication default credentials, Development status, Has changed, Import type, JMS provider, JMS transport, JNDI initial context factory, Keep up to date, Messaging model, Override address, Proxy default credentials, Toolkit, Use authentication, Use proxy, and Visibility.

The following **Progress** graph shows the time interval of the application components as a function of the total number of components. The colored lines indicate the various types of application components: General, JMS, Web Service, and so forth.

Trend Graphs
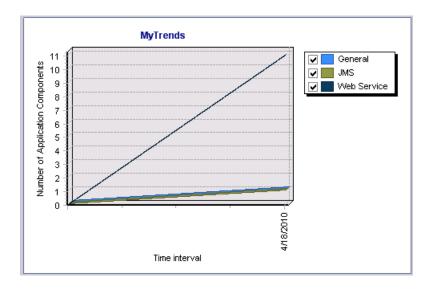
You can set up a **Trend** graph to plot the trend of application components against a time interval.

You can configure the time period to: Last 7 days, Last 5 weeks, Last 12 months, or All days.

You can configure the grouping to: All fields, Assigned to Dev, Assigned to QA, Authentication default credentials, Development status, Has changed, Import type, JMS provider, JMS transport, JNDI initial context factory, Keep up to date, Messaging model, Override address, Proxy default credentials, Toolkit, Use authentication, Use proxy, and Visibility.

The following **Trend** graph shows the different component types as a function of their creation dates.

Application Changes Graphs

You can create graphs to plot the application's changes using Service Test Management's built-in graphs. You can create **Progress**, **Summary**, and **Trend** graphs for the application components and for the application changes.

Summary Graphs
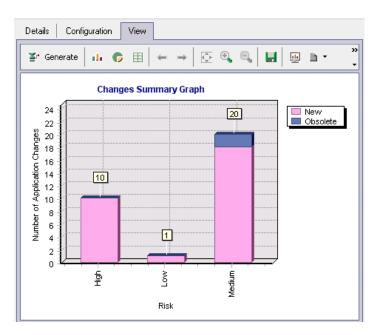
In Application Changes Summary graphs, you can configure the **x-axis** to display one the following fields: Affected part type, Assigned to, Change status, Change type, Created by, Cycle, Release, Risk, and QC Project.

The **y-axis** display the number of (**Count)** of the application changes.

You can filter the graph by all of the fields, including all system and UDF fields defined for the entity, such as Affected part type, Assigned to, Change ID, Change status, Change type, Created by, Creation date, Creation time, Cycle, Release, Risk, Subject, or custom fields.

You can group the results by the following criteria: Affected part, Affected part type, Assigned to, Change status, Change type, Created by, Creation date, Creation time, Cycle, Release, Risk, or QC Project.

The following **Changes Summary** graph shows the number of risks, grouped by the risk level, as a function of the total number of changes.

Progress Graphs

You can set up a **Progress** graph to plot the number of application changes against a time interval.

You can configure the **x-axis** to display a time interval: Last 7 days, Last 5 weeks, Last 12 months, or All days.

The **y-axis** display the number of (**Count)** of the application changes.

You can group the results by the following criteria: Assigned to, Change status, Cycle, Release, or Risk.

The following **Changes Progress** graph shows the time interval of the application components as a function of the total number of components. The colored lines indicate the various levels of risk: **High**, **Low**, and **Medium**.

Trend Graphs

You can set up a **Trend** graph to plot the trend of application changes against a time interval.

You can configure the time period to: Last 7 days, Last 5 weeks, Last 12 months, or All days.

You can configure the grouping to: All fields, Assigned to, Change status, Cycle, Release, and Risk.

Tasks

🐦 How to Create and Generate Reports

This task describes how to create reports using ALM's built-in reports for application components.
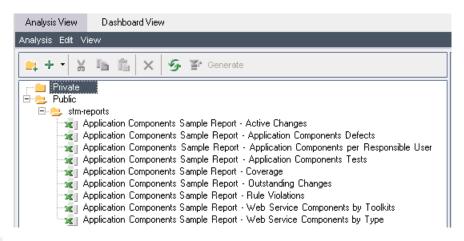
This task includes the following steps:

➤ "Generate a pre-defined report" on page 225

➤ "Create a custom report based on a pre-defined report" on page 226

➤ "Create new Excel reports" on page 226

Generate a pre-defined report

Use the **Dashboard** module to generate pre-defined Excel reports.

1 Open the **Dashboard** module and select the **Analysis View** tab.

2 Select a report from the **Public/stm-reports** folder.

3 Click **Generate**. Specify a location for the Excel report.

Create a custom report based on a pre-defined report

You can use an existing report as a basis for creating a custom report.

1 Open the **Dashboard** module and click the **Analysis View** tab.

2 Select a report from the **Public/stm-reports** folder.

3 To customize a built-in report without providing access to other users, copy it to the **Private** folder and rename it.

To customize a built-in reports and allow everyone to access your report, keep it in the Public folder.

4 Open the **Configuration** tab in the right pane, and click the **Query Builder** button.

5 Select an entity in the right pane's **Entities View** and expand it.

6 Drag the desired parameter to a location within the SQL statement. Modify the SQL as required.

7 Click **OK** to save it and close the **Query Builder**.

8 Click **Generate** and specify a location for the Excel report.

Create new Excel reports

You can design a new report without basing it on a pre-defined one.

1 Open the **Dashboard** module and click **Analysis View.**

2 Click on the target folder and subfolder:

➤ **Private**: Report is only available to current user.

➤ **Public**: Report is available to all users of the project.

3 Choose **Analysis** > **New Excel Report**. Specify a name for the report.

4 Click **Query Builder**. ALM opens the **Entity View** in the right pane.

5 Select an entity in the right pane and expand it to view the parameters. For a list of entities, see "Application Component Query Builder Entities" on page 218.

6 Drag a parameter to a location within the Query Builder editor. Modify the SQL as required.

7 Use the **Test Query** and **Run Query** button to test and run the query.

8 Click **OK** to save the report. You can open this report and run the query at a later time.

For details on how to customize the Excel reports, see the *HP ALM User Guide*.

🐦 How to Generate Graphs

This task describes how to create graphs for your Application Components and the changes associated with them.
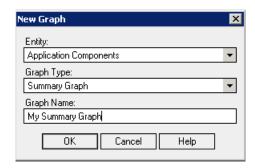
This task includes the following steps:

➤ "Open the New Graph dialog box" on page 227
➤ "Configure the graph" on page 228
➤ "Select different projects - optional" on page 228
➤ "View the Graph" on page 228

Open the New Graph dialog box

Use the **Dashboard** module to generate pre-defined graphs.
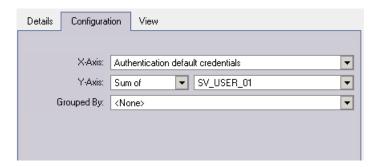
1 Open the **Dashboard** module and select the **Analysis View** tab.

2 Select **Analysis > New Graph (Ctrl+G)**.

3 Select an Entity - **Application Components** or **Application Changes**.

4 Select a graph type: **Summary Graph**, **Progress Graph**, or **Trend Graph**.

5 Specify a graph name and click **OK**.

Configure the graph

1 Click the **Configuration** tab in the right pane.

2 Select values for the relevant fields.

For details about the configuration options, see "Application Component Graphs" on page 219 or "Application Changes Graphs" on page 222.

Select different projects - optional

To plot graphs using data from other projects, click **Select Projects** in the bottom right pane, and select the relevant project.

View the Graph

Click the **View** tab to view the graph.

For more information on graphs, see the *HP ALM User Guide*.

Index

J

JMS test settings 54
JNDIU definitions 54

K

Knowledge Base 17

L

Link Test to Requirement 152
linking to requirement, tests 148
linking, components to tests 149
longevity testing 108

M

mailing services 95
Manual Runner, testing services 83
Modeling tab 43
module access 191

N

New Change dialog box 204
New Component dialog box 51
New Folder dialog box 50
New Group dialog box 103

O

operations
    coverage 140
    coverage overview 123
    coverage, viewing 128
Operations tab 140
Overload testing 108
Owner, Changes tab 208

P

Part Name, Change tab 205
Part Type, Change tab 205
Performance testing
    longevity 108
    overload 108
    scalability 108

stress 108
    volume 108
pivot table 218
Positive testing 107

Q

Query builder, for Excel report 218

R

rage 121
registry, Systinet 21
reports, generate and create 225
required fields 184
requirements
    coverage 137
    coverage overview 122
    coverage, checking 128
Requirements tab 137
rules
    coverage 133
    violations 120
Rules tab 133, 175, 179
Run operation dialog box 80

S

Scalability testing 108
search, for components 91
Security testing
    cross-site scripting 108
    SQL injection vulnerability 108
Service Test Management, overview 20
Service Test, creating scripts 67
spot testing 66
SQL injection vulnerability 108
Standard Compliance testing 107
Stress testing 108
Systinet registry 21
Systinet, adding from 72

T

Test Plan module
    selecting components 158
Test Plan module, components 156

Index