# **HP Service Test Management**

Software Version: 10.50

# User's Guide

Document Release Date: May 2010 Software Release Date: May 2010



# **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.

#### Third-Party Web Sites

HP provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. HP makes no representations or warranties whatsoever as to site content or availability.

### Copyright Notices

© Copyright 2000 - 2010 Hewlett-Packard Development Company, L.P.

#### Trademark Notices

Java<sup>TM</sup> is a US trademark of Sun Microsystems, Inc.

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**

Service Test Management User Guide	9
Chapter 1: Introduction to Service Test Management	15
Concepts	15
Service Test Management Overview	15
Chapter 2: Creating Application Components	19
Concepts	19
Creating Application Components Overview	19
Tasks	24
How to Define Components	24
Reference	26
Quality Center Direct Links	
Chapter 3: Web Services	55
Concepts	56
Defining Web Services Overview	
Tasks	60
How to Define a Web Service	60
Reference	63
Web Service Specific User Interfaces	63

Chapter 4: Managing Components in Quality Center	77
Concepts	77
Managing Components	77
Tasks	80
How to Manage Components	80
Chapter 5: Groups	83
Concepts	83
Groups Overview	83
Tasks	85
How to Manage Component Groups	85
Reference	87
New Groups User Interface	87
Chapter 6: Generating Requirements and Tests	89
Concepts	89
Generating Requirements and Tests Overview	89
Tasks	
How to Generate Requirements and Tests	94
Reference	
Generate Requirements and Tests User Interface	95
Chapter 7: Determining Test Coverage	103
Concepts	104
Coverage Overview	104 105 106 107
Tasks	109
How to Check the Testing Status	109

Reference	112
Testing Status Tab	112
Chapter 8: Linking Components and Requirements	123
Concepts	123
Linking Components Overview	123
Tasks	124
How to Link Components to Requirements and Tests	124
Reference	125
Linking Component User Interface	125
Chapter 9: Customizing Service Test Management	133
Concepts	133
Customization Overview	133
Tasks	138
How to Customize a Project	138
Reference	140
Project Customization User Interface	140
Chapter 10: Change Analysis	155
Concepts	155
Change Analysis Overview	155
Tasks	158
How to Perform Change Analysis	158
Reference	161
Changes Tab	161
Chapter 11: Reports and Graphs	1 <b>7</b> 1
Concepts	171
Service Test Management Reports and Graphs Overview	171
Tasks	181
How to Create and Generate Reports	

### **Table of Contents**

Chapter 12: Custom Script Generators	185
Concepts	185
Creating Custom Script Generator Overview	185
Tasks	186
Implement and Register a Custom Generator	186
Reference	189
Script Generator API Reference	189
Test Generation Flow	192
Index	195

# **Service Test Management User 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 Quality Center 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 Quality Center. For additional information on using Quality Center, refer to the *HP Quality Center User's Guide*.

# **How This Guide Is Organized**

This guide contains the following chapters:

# **Chapter 1** Introduction to Service Test Management

Provides an introduction to the Service Test Management model and its integration with Quality Center.

# **Chapter 2** Creating Application Components

Describes how to create a Components tree and define components and component groups.

#### **Chapter 3** Web Services

Describes how to work with Web Service type application components and the unique features available for Web Services.

### **Chapter 4** Managing Components in Quality Center

Describes how to view and modify component details, definitions, and operation data, generate Excel reports, remove and restore components, and define component dependencies.

### **Chapter 5** Groups

Describes how to create component groups and associate application components to them.

# **Chapter 6** Generating Requirements and Tests

Describes how to use the Requirement and Test Generation wizard to automatically generate requirements and tests and how to create empty tests.

# **Chapter 7 Determining Test Coverage**

Shows the testing status with graphical displays of test, requirement, and defect coverage.

# **Chapter 8** Linking Components and Requirements

Describes how to link components to requirements and tests.

# **Chapter 9** Customizing Service Test Management

Describes how to customize the Application Components module, user group permissions, module access, and project entities for components and groups.

# **Chapter 10 Change Analysis**

Describes how to identify changes in components and analyze their impact on related assets in Quality Center. Also describes how to create and run test sets to test the impact of these changes.

### **Chapter 11 Reports and Graphs**

Describes how to generate graphs and Excel reports to represent your components and their coverage.

# **Chapter 12 Custom Script Generators**

Extends the capabilities of the Service Test Management Requirement and Test Generation Wizard through an external script generator.

# **Documentation Library**

The Documentation Library is an online help system that describes how to use the application. When using Service Test Management in the Quality Center application, you can access the Quality Center Documentation Library or the Service Test Management Documentation Library.

- ➤ Quality Center Documentation Library. When working in any module other than the Application Components module, you can access the Quality Center Documentation Library in the following ways:
  - ➤ Click **Documentation Library** in the Help menu to open the Quality Center Documentation Library home page. The home page provides links to the Quality Center guides and references.
  - ➤ Click **Help on this page** in the Help menu to open the Documentation Library to the topic that describes the current page.
- > Service Test Management Documentation Library. When working in the Application Components module, you can access the Service Test Management Documentation Library. Select Help > Help on this page to open the Help window. Click the Search and Navigate button in the top left corner and then go to the Home link. The Service Test Management Documentation Library home page provides links to the Testing guides.

The Documentation Library is best viewed from a browser with Java support. If you do not have Java support on your browser, download the Sun Java plug in from the Sun Java Web site (<a href="http://java.com/en/index.jsp">http://java.com/en/index.jsp</a>). Note that if Java support is not available, the Documentation Library automatically opens using the JavaScript implementation. The JavaScript implementation provides the same basic functionality as the Java implementation, however does not allow use of the Favorites tab within the navigation pane.

### **Service Test Management Documentation Library Guides**

The Service Test Management Documentation Library consists of the following guides, 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).

**Readme** provides last-minute news and information about Service Test Management.

**HP Service Test Management User's 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 Service Test Management Installation Guide** explains how to install Service Test Management on a server machine as an extension to Quality Center. (Available in PDF format only.)

**HP Service Test User's Guide** explains how to use Service Test to create scripts for SOA environments. (Available in PDF format only.)

### **Quality Center Documentation Library Guides**

The Quality Center Documentation Library consists of the following guides and references, 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).

**Getting Started** explains how to use the Documentation Library and how it is organized. (Available online.)

**What's New?** describes the newest features in the latest versions of Quality Center. (Available online and in PDF format.)

You can also access the What's New? from the Quality Center **Help** menu. In addition, you can choose **Help** > **Product Feature Movies** to view short movies which demonstrate the main product features.

**Readme** provides last-minute news and information about Quality Center.

#### **Quality Center Guides**

**HP Quality Center User's Guide** explains how to use Quality Center to organize and execute all phases of the testing process. It describes how to define requirements, plan tests, run tests, and track defects. (Available online and in PDF format.)

**HP Quality Center Administrator's Guide** explains how to create and maintain projects using Site Administration, and how to customize projects using Project Customization. (Available online and in PDF format.)

**HP Quality Center Tutorial** is a self-paced guide teaching you how to use Quality Center to manage the software testing process. (Available in PDF format.)

**HP Quality Center Installation Guide** explains how to install Quality Center on a server machine in a cluster environment or as a stand-alone application. (Available in PDF format.)

#### **Business Process Testing Guides**

**HP Business Process Testing User's Guide** explains how to use Business Process Testing to create business process tests. (Available online and in PDF format.)

**HP Business Process Testing Tutorial** provides a self-paced guide that teaches you the basics of Business Process Testing in the Quality Center application. (Available in PDF format.)

#### **API References**

**HP Quality Center Database Reference** provides a complete online reference for the project database tables and fields. (Available online.)

**HP Quality Center Open Test Architecture API Reference** provides a complete online reference for the Quality Center COM-based API. You can use the Quality Center open test architecture to integrate your own configuration management, defect tracking, and home-grown testing tools with a Quality Center project. (Available online.)

#### Welcome to This Guide

**HP Quality Center 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 Quality Center users, projects, domains, connections, and site configuration parameters. (Available online.)

**HP Quality Center Custom Test Type Guide** provides a complete online guide on how to create your own testing tool and integrate it into the Quality Center environment. (Available online.)

# **Introduction to Service Test Management**

This chapter includes:

#### Concepts

➤ Service Test Management Overview on page 15

# 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 Quality Center, enables you to centrally manage your assets. Using this module, you can define and store application components in Quality Center'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 16
- ➤ "The Advantages of Service Test Management" on page 16

# 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 Quality Center.** Enables you to centrally manage SOA assets from the Application Components module. You can define, import, and store services in Quality Center, 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 services, and create an emulation of a Web service for testing purposes.
- ➤ 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.

**Chapter 1 •** Introduction to Service Test Management

# **Creating Application Components**

#### This chapter includes:

#### **Concepts**

➤ Creating Application Components Overview on page 19

#### **Tasks**

➤ How to Define Components on page 24

#### Reference

- ➤ Quality Center Direct Links on page 26
- ➤ Application Components Module User Interface on page 27

# **Concepts**



# Creating Application Components Overview

You add entities to the Application Components tree by manually adding a new components. After you add a component, you can describe it through a contract.

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

This section also includes:

➤ "The Application Component Tree Hierarchy" on page 23



# 🔍 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 Quality Center 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 Quality Center. For details about the Obsolete folder, see "Remove and restore components - optional" on page 82.

You cannot rename, move, or delete the **Application Components** folder or the **Obsolete** folder.

### 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.

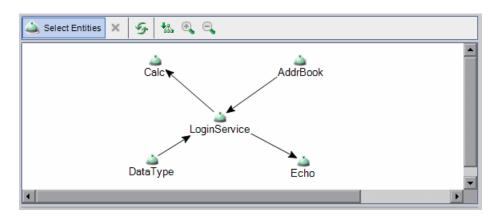
You can define the dependencies using Service Test Management's **Modeling** tab. The **Modeling** tab displays a diagram of the component dependencies.

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.

#### **Chapter 2 • Creating Application Components**

You select components and add them to a modeling diagram. In the following example, the **LoginService** component is dependent on **AddrBook** and **DataType**. The **Echo** and **Calc** components are dependent on **LoginService**.



Although you drag in the dependent components, the dependencies are actually per operation. For details, see "Define dependencies - optional" on page 25.

# **Tasks**

# **P** 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 25
- ➤ "Create a new component" on page 25
- ➤ "Specify a contract" on page 25

#### **Create a folder - optional**

Open the New Folder dialog box and add new folders and subfolders to create a tree hierarchy for your components. For details, see the "New Folder Dialog Box" on page 43.

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 and provide whatever details are available to you. For details, see the "New Application Component Dialog Box" on page 43.

### 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 43.

### **Define dependencies - optional**

- **1** Select the **Modeling** tab. For a description, see the "Modeling Tab" on page 36.
- **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.
  - 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 42

# Reference

# 🔍 Quality Center 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>&amp;Uui d=<uuid></uuid></environment>

View	Description
Component	StmModule00000000000003130674279? EntityType=IStmService&EntityID= <component id="">&amp;View=<service tab="" type=""></service></component>
	<component tab="" type=""> values:</component>
	<ul> <li>Changes</li> <li>Modeling</li> <li>TestingStatus</li> <li>AspectStatus</li> <li>OperationStatus</li> <li>StatusDashboard</li> </ul>
Component Group	StmModule0000000000003130674279? EntityType=IStmGroup&EntityID= <group id="">&amp;View=<group tab="" type=""></group></group>
	<pre><group tab="" type=""> values:</group></pre>
	➤ Modeling
	➤ TestingStatus
	➤ AspectStatus
	➤ StatusDashboard

# Application Components Module User Interface

Service Test Management provides a user interface to organize your application components and customize the Quality Center project hosting them.

This section includes (in alphabetical order):

- ➤ Alerts Dialog Box on page 28
- ➤ Details Tab on page 29
- ➤ Interaction Tab on page 32
- ➤ Find Dialog Box on page 35
- ➤ Modeling Tab on page 36

#### **Chapter 2 •** Creating Application Components

- ➤ Dependency Details Dialog Box on page 42
- ➤ New Folder Dialog Box on page 43
- ➤ New Application Component Dialog Box on page 43
- ➤ Application Components Module Icons on page 47
- ➤ Application Components Module Window on page 48
- ➤ Application Components Module Menus and Buttons on page 50

# Alerts 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 <b>Application Components</b> tree.
Relevant tasks	"Evaluate the tests" on page 160
See also	"Change Analysis" on page 155

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

UI Elements (A-Z)	Description
×	Removes the current rule violation in the <b>Rule Violations</b> pane:
	Note: When removing a rule violation, Service Test Management disregards it—it does not delete it.
× Clear All	Removes all rule violations in the <b>Rule Violations</b> pane: <b>Note:</b> When removing a rule violations, Service Test  Management disregards it—it does not delete it.

UI Elements (A-Z)	Description
>> Show	For change alerts, this button opens the <b>Changes</b> tab listing all of the changes. Use the <b>Create Test Set</b> utility to resolve the changes.
	For details, see the "Create Change Impact Test Set Wizard" on page 167
>> Resolve	For rule violations, this button opens the <b>Create Requirements Test</b> wizard in order to resolve the violation. For details, see the "Generate Requirements and Tests Wizard" on page 97.
Alerts for <component name=""></component>	A link to the component in the <b>Application Components</b> 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.

# **Q** Details Tab

Displays and enables you to edit a component's details.

To access	Select <b>Application Components</b> module, highlight a folder, component, or group, and click the <b>Details</b> 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 80
See also	"Application Component Details Pane" on page 39

# **Details Tab for Components**

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

GUI Element (A-Z)	Description
<user-defined></user-defined>	User-defined fields defined in customization.
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 as provided by the user.
Component Type	The type of component. The built-in components are <b>General, Web Service</b> , and <b>JMS</b> . 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.
<b>Creation Date</b>	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: <b>Y</b> or <b>N</b> . When enabled, instructs Service Test Management to update the component from its source each time you select it in the Application Component's tree.
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 **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.
<b>Creation Date</b>	The date the group was created in the <b>Components</b> 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 **Components** tree:

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

# **1** Interaction Tab

Displays and enables you to edit a component's contract information.

To access	Select <b>Application Components</b> module, highlight a component, and click the <b>Interaction</b> 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 80
See also	"Application Component Details Pane" on page 39.

The Interaction tab contains the following information:

UI Elements (A-Z)	Description
Contract Details	<ul> <li>Contract File. The location of the contract file defining the application component.</li> <li>Browse</li></ul>
Contract Details - Web Services	<ul> <li>Import type. Contract import method: File, URL, Systinet, and UDDI.</li> <li>Toolkit. The toolkit associated with the Web service. You set this in the Customization settings, and it is read-only.</li> <li>Service Address. The service deployment location, to which service requests are sent. By default, Quality Center displays the endpoint address specified in the WSDL file.</li> <li>Override Address. Enables you to enter an alternate endpoint for the service in the Service Address box.</li> </ul>

**Chapter 2 •** Creating Application Components

UI Elements (A-Z)	Description
Contract Details - JMS	<ul> <li>JMS connection factory. The JNDI name of the JMS connection factory. This setting is unique per test.</li> <li>JMS destination. The name of the JMS destination—either the queue or topic name.</li> <li>JMS provider. The JMS service provider: Apache ActiveMQ, JBoss Messaging from JBoss, Open Message Queue, Websphere MQ from IBM, and so forth.</li> <li>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.</li> <li>JNDI provider URL. The URL of the service provider. For example: Websphere - iiop://myserver:myport</li> <li>Messaging Model. The type of JMS message—Point to</li> </ul>
Interaction Fields	point, or Publish and subscribe.  The fields you designated in the Customization for the component type. For details, see "Component Types Tab" on page 147.  The common fields are:  > JMS Request Queue  > JMS Response Queue  > JMS Transport (Y or N)  > WSDL Native Name
Operations	<ul> <li>Refresh. Reloads the list of operations.</li> <li>Web Service Call         . 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 Service Testing Tab" on page 71. Note: This option is only available if Service Test or the Service Test add-in is installed on the machine.     </operation-name></li> <li><operation list="">. A list of the available operations for the application component.     </operation></li> </ul>
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

Enables you to search for a component, group, or folder within the Components tree.

To access	Edit > Find
Relevant tasks	"Find folders and components - optional" on page 81
See also	"Manage component details - optional" on page 80

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.

#### **Chapter 2 •** Creating Application Components

UI Elements (A-Z)	Description
Entity Type	Type of entity in which to perform the search: Folder, 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 29.
Match Criteria	Advanced match criteria:
	➤ Case Sensitive. Performs a case sensitive search.
	➤ Exact Match. Looks for the exact string—not a partial string.
	➤ <b>Use Wildcard</b> - Applies wildcard rules to the search text.
Value to Find	The text to find in the component, group, or folder.

# Nodeling 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	Choose the <b>Application Components</b> module, select a component or group, and click the <b>Modeling</b> tab.
Important information	If you select a component, the visual design shows the component dependencies.
Relevant tasks	<ul><li>"Define dependencies - optional" on page 25</li><li>"Add a component to a group" on page 86</li></ul>
See also	"Component Dependencies" on page 23

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

UI Elements (A-Z)	Description
- Select	Opens the <b>Components</b> tree in the right pane to add more component or groups to the modeling window. For information about the tree hierarchy, see the "Components Tree" on page 38.
×	<b>Delete.</b> Removes the selected entity from the visual area. You cannot delete the base component or group that is selected in the left pane's <b>Component</b> tree.
5	<b>Refresh.</b> Redraws the visual diagram in the visual area.
<b>*</b> 35	<b>Arrange layout</b> . Arranges the layout of all entities in the current modeling window, in a symmetric fashion.
⊕_	<b>Zoom in.</b> Enlarges the entities in the current drawing.
<b>Q</b>	<b>Zoom out.</b> Reduces the size of the entities in the current drawing.
<modeling window=""></modeling>	A visual area showing component dependencies or component/group relationships in a graphical representation.
Application Component/Group	Details of the component or group selected in the modeling window.
Details	➤ For component details, see "Application Component Details Pane" on page 39
	➤ For group details, see "Group Details" on page 41

### **Components Tree**

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

UI Elements (A-Z)	Description
4-	Add. Opens a dropdown list with the following options:
,	➤ <b>Associate.</b> Associates the selected component with a group.
	➤ Add Dependency (Dependency To)
	➤ Add Dependency (Dependency From)
	<b>Note:</b> The <b>Associate</b> option is enabled when you select a <i>group</i> in the right pane. The <b>Add Dependency</b> options are enabled when you select a <i>component</i> in the right pane.
	For details about dependencies, see the "Dependency Details Dialog Box" on page 42
5	Refresh. Refreshes the Components list.
7	<b>Set/Clear Filter.</b> Opens a dropdown 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.
×	<b>Exit.</b> Closes the Components tree pane
	To reopen this view, click the Select  button in the <b>Modeling</b> tab
<components list=""></components>	An expandable list of all components by folder and group.
	<b>Tip:</b> If a filter is active, there could be hidden components. To see all components, click <b>Clear Filter</b> .

### **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>	User-defined fields defined in customization.
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 Folder Path	The folder path of the component in the main Components tree.
Component ID	A read-only ID assigned to the component.
Component Name	The native component name in the definition files.
Component Type	The type of component: <b>General, Web Service</b> , and <b>JMS</b> . If you defined a new component type in Customization, it will be displayed.
Contract file	The path or URL of the component's contract file.
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 <b>Application Components</b> tree.
Deleted On	The date the component was deleted. This applies primarily to components in the <b>Obsolete</b> folder.
Development Status	The development status of the component:  Development, Maintenance, QA ready, or Stable.
Has Changed	Y or N. Yes indicates that the definition files (for example, WSDL), changed since the last update.
Import type	Contract import method: File, URL, Systinet, or UDDI.

### **Chapter 2 •** Creating Application Components

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.
Location	The location of the component's source file (such as WSDL) that is displayed by default when defining the component. To override the default, select <b>Override Address</b> .
Original Location	The original location of the WSDL This is relevant when you choose <b>Override address</b> and change the endpoint.
Override Address	Enables you to enter an alternate endpoint for the service in the <b>Service Address</b> box (Web Services only).
Port Type	The port type of the proxy server.
Service Address	The service deployment location, to which requests are sent. By default, Quality Center displays the endpoint address specified in the WSDL file. (For Web Service type components only)
Systinet ID	The Systinet ID for the imported Web services.
Systinet Key	The Systinet key for the imported Web services.
Systinet Server	The URL address of the Systinet server from which the service definition is imported.
Systinet Server Port	The port of the Systinet server machine.
Systinet Version	The version of the Systinet server hosting the service.
Toolkit	The toolkit associated with a Web service. You set this in the Customization settings, and it is read-only.
UDDI Server	The URL address and version of the UDDI server from which the service definition is imported.
UDDI Service Key	A unique identifier of the service on the UDDI server, used to locate the service definition when updating the service.
UDDI Version	The version of the UDDI registry: <b>2</b> or <b>3</b> .

UI Elements (A-Z)	Description
<b>Update Time</b>	The time the WSDL was updated.
Visibility	The visibility of the application component: <b>Internal</b> or <b>External</b> .
WSDL native name	The name of the Web Service as determined from the WSDL file. (For Web Service type components only)

### **Group Details**

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.

# Dependency Details Dialog Box

Enables you to set the dependencies between components or modify existing ones.

To access	In Modeling tab
	➤ For a new dependency: 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 grid, select a dependency and click the Edit button ■.
Important	An arrow indicates a dependency:
information	➤ 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 <b>Dependency To</b> another component—the added component is impacted by the base component.
Relevant tasks	"Define dependencies - optional" on page 25

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

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

# New Folder Dialog Box

This dialog box page enables you to create a new folder in the Components tree.

To access	Use one of the following:  ➤ Click the New Folder   ➤ 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 25

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 Components tree. You can create several levels of sub-folders.
	<b>Tip:</b> Add a description to the folder in the <b>Details</b> tab.
ОК	Adds the new folder to the Components tree.

# New Application Component 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 25

### **Chapter 2 •** Creating Application Components

The following elements are included:

II. Clears all the fields so that you can redefine the nent.  pane. The component's properties. The displayed affer based on the component type and the fields
iffer based on the component type and the fields
in the customization section. For details, see the onent Types Tab" on page 147. Some common ee:
esigned to Dev esigned to QA eated by evelopment Status: Development, Maintenance, A ready, or Stable. st Updated

UI Elements (A-Z)	Description		
-0:	Interaction pane. Contract information.		
<b>₹</b>	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 <b>Interaction Field</b> column. For details, see the "Component Types Tab" on page 147.		
	<ul> <li>Contract File. The location of the contract.</li> <li>Browse</li> <li>For non-Web Service components, opens the Open</li> </ul>		
	<ul> <li>File dialog box.</li> <li>For Web Services, opens the Import Service dialog box. For details, see "Import Service Dialog Box" on page 68.</li> </ul>		
	➤ <b>Operations.</b> 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).		
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.		
Ci,	Attachments pane. Allows you to attach a file to the component. For details, see the <i>Quality Center User Guide</i> .		

### **Chapter 2 •** Creating Application Components

UI Elements (A-Z)	Description
Submit	<b>Submit</b> . Creates the component whose details you specified.
Close	<b>Close.</b> Closes the dialog box and cancels any changes you made to the existing screen.
Component Name	A name for the component.
Component Type	The type of component to create: <b>General</b> , <b>Web Service</b> , or <b>JMS</b> . If you created a custom type component in Customization, it also appears in the dropdown list.

# Application Components Module Icons

This section describes the icons available in the **Application Components** module.

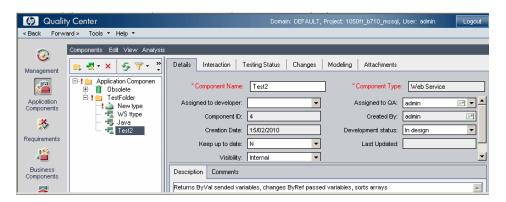
To access	Click the <b>Application Components</b> button quality Center sidebar.
See also	<ul> <li>"Application Components Module Window" on page 48</li> <li>"Application Components Module Menus and Buttons" on page 50</li> </ul>

The icons are described below:

UI Elements	Description
1	<b>Alerts.</b> Click to view a summary of the following alert types:
	<ul> <li>Changes. The component definition has changed.</li> <li>Rule Violations. The component violates a rule.</li> </ul>
	For more details, see the "Alerts Dialog Box" on page 28.
	Folder. A folder for organizing components.
Ŭ	<b>Obsolete folder.</b> Contains components that were deleted that may affect other components.
·m	General type. A General type application component.
-T	<b>Web Service type.</b> A Web Service type application component.
<b>*</b>	JMS type. A JMS type application component.
	Application Component group.

# Nation Components Module Window

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



To access	Click the <b>Application Components</b> button Quality Center sidebar.
Relevant tasks	"Manage component details - optional" on page 80
See also	<ul> <li>"Application Components Module Icons" on page 47</li> <li>"Application Components Module Menus and Buttons" on page 50</li> </ul>

User interface elements are described below:

UI Elements (A-Z)	Description
<toolbar and="" menu=""></toolbar>	Provides menus and toolbar buttons specific for the <b>Application Components</b> module. For details, see the "Application Components Module Menus and Buttons" on page 50.
<components tree=""></components>	Shows a tree hierarchy of all the components, component groups, and folders. For details, see "Application Components Module Icons" on page 47.

UI Elements (A-Z)	Description	
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 Quality Center 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 29.	
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 157.	
Interaction tab	Provides the component's contract details, such as the WSDL location for Web Services. For details, see the "Interaction Tab" on page 32.	
Modeling tab	Provides an interface to graphically associate components with groups, and to define dependencies between them. For details, see "Modeling Tab" on page 36.	
Testing Status tab	Provides a coverage summary for aspects, requirements, operations, tests, rule violations, and defects. For details, see "Determining Test Coverage" on page 103.	

# Application Components Module Menus and Buttons

This section describes the menus and buttons available in the Requirements module.

To access	Click the <b>Application Components</b> button on the Quality Center sidebar.
Important information	All menu items are available from the right-click shortcut menu.
Relevant tasks	"Manage component details - optional" on page 80

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

UI Elements (A-Z)	Menu	Description
=1	Toolbar	New Folder. Opens the New Folder dialog box to create a folder under the root or an existing folder.
er -	Toolbar	New Component. Displays the last button used: New Component or New Group.
属	Toolbar	New Group. Displays the last button used: New Component or New Group.
×	Toolbar	Delete. Deletes selected components. Deleting a component also deletes its children. You cannot delete the root folder.
5	Toolbar	Refresh All. Refreshes the Components tree so that it displays the most up-to-date versions.
▼・	Toolbar	Filter/Sort. Enables you to filter and sort the requirements in the requirements tree or grid. For more details, see below.

UI Elements (A-Z)	Menu	Description
<u></u>	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 Quality Center 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 Quality Center and takes you to the requirement. If you are not already logged in, Quality Center 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:
		<ul> <li>Coverage and modeling data for the component are not copied.</li> <li>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.</li> </ul>
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.

### **Chapter 2 •** Creating Application Components

UI Elements (A-Z)	Menu	Description
Cut/Paste	Edit	Moves selected component to a different location in the Components tree.
		<b>Tip:</b> You can also move a component to a new location by dragging it.
× Delete	Edit	Deletes the selected component.
<b>Download Contract File</b>	Components	Allows you to download a contract file from Quality Center to a local file system.
Emulation	Components	Provides access to the Web service emulation tools:
		➤ Emulate Service. Opens the Specify ServiceEmulation Host dialog box.
		<ul> <li>Run Service Emulation Console.</li> <li>Opens the console enabling you to run the service.</li> </ul>
Expand All/Collapse	View	Expands all or collapses the selected branches in the Component tree.
Filter/Sort	View	Enables you to filter and sort application components in the tree. For more details, click the <b>Help</b> button.
Find	Edit	Searches for a component in the <b>Application Components</b> module. For details, see "Find Dialog Box" on page 35.
Generate Requirements/ Test	Components	Opens the Generate Requirements and Test wizard. For details, see "Generating Requirements and Tests" on page 89.

**Chapter 2 •** Creating Application Components

UI Elements (A-Z)	Menu	Description
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.

### **Chapter 2 •** Creating Application Components

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 <b>Copy URL</b> . For more details, see the <i>HP Quality Center User Guide</i> .
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 56
- ➤ Creating Tests in Service Test on page 59

#### **Tasks**

➤ How to Define a Web Service on page 60

#### Reference

➤ Web Service Specific User Interfaces on page 63

### 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 to Web Services, such as Spot Testing, Service Emulation, and Script Creation with HP Service Test.

When working with Web Services, you create a Component hierarchy as described in "Creating Application Components Overview" on page 19.

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 Quality Center repository.

For defining application components that are not WSDL-based, see the "New Application Component Dialog Box" on page 43.

If your WSDL is not fully developed, or if the service is currently unavailable, you can test with an emulated service. For details, see "Create an emulated Web service - optional" on page 61.

This section also includes:

- ➤ "Importing Services" on page 57
- ➤ "Secure Services and Proxy Servers" on page 57
- ➤ "Spot Testing Operations" on page 58



### 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 Quality Center repository.

You can then link the service to other Quality Center 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 89.

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 57.

For user interface details, see the "Import Service Dialog Box" on page 68.

### 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 133.



### 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.

#### **Chapter 3 •** Web Services

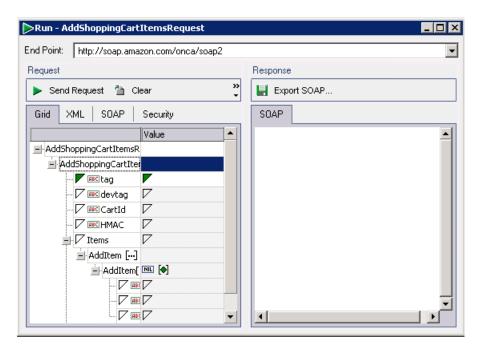
The security or proxy information is always associated with the WSDL, If you enable the **Keep up to date** option to allow automatic synchronization, Service Test Management uses the authentication or proxy server settings to access the WSDL at its source.

**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 138.



### 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.



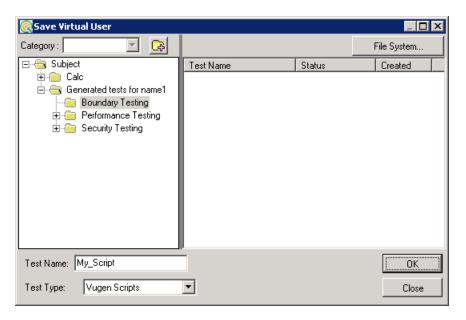
You use the grid to enter request values and submit them to the server. You can view the SOAP response and export it to a file. For details, see "Test the service - optional" on page 62.

# 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, use the **Tools** menu to connect to the same Quality Center project that contains your Web Service type components.

Create a Web Services type script in the normal way, and save it to the appropriate location in the Quality Center repository.



Use the **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 130.

For more information, see the *HP Service Test User Guide*, available from the online help's Home page.

### **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 60
- "Create a new component" on page 60
- "Import a service" on page 61
- ➤ "Set authentication or proxy information optional" on page 61
- "Create dependencies optional" on page 61
- ➤ "Create an emulated Web service optional" on page 61
- "Test the service optional" on page 62

#### 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 43.

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

### Create a new component

Create a new component. Select **Components > New > New Component**. For details, see the "New Application Component Dialog Box" on page 43.

#### **Import a service**

In the New Component Dialog box, select the **Interaction** link and click **Browse** to open the Import Service dialog box. For details, see the "Import Service Dialog Box" on page 68.

#### 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 66.

#### **Create dependencies - optional**

Dependencies indicate the reliance of one component upon another.

- **1** Select the **Modeling** tab. For a description, see the "Modeling Tab" on page 36.
- **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)**.
  - **4** 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 42.

### **Create an emulated Web service - optional**

If you are working with a Web service component that is unavailable or still under development, you can create an emulation of the Web service component. This requires an installation of HP Service Test or the Service Test Add-in for Quality Center.

- 1 Verify that the host server is active. The host server is the machine to which you will be sending the Web Service calls. By default, Service Test activates the server upon startup. Enter the following URL into your browser:
  - http://<hostname>:8080/ServiceEmulation/index.jsp.
    If the server is active, the browser will display HP Service Emulation.
- 2 If the server is not active, choose Start > All Programs > HP Service Test > Start Emulation Service.
- **3** Select **Components** > **Emulation** > **Emulate Service**. In the Specify Service Emulation Host dialog box, specify the machine to which you will be sending the Web Service calls. This can also be the local machine.
- **4** Set up the service emulation. Select **Components** > **Emulation** > **Run Service Emulation Console**. Configure the rules as required, For more information, see the *HP Service Test User Guide*, available through the online help's **Home** page.

#### 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 69.

#### Run the test

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

The manual running option provides an interface for working with Web Services. For details, see the "Manual Runner Service Testing Tab" on page 71.

### 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 63
- ➤ Select Service from UDDI Dialog Box on page 64
- ➤ Connection Settings Dialog Box on page 66
- ➤ Import Service Dialog Box on page 68
- ➤ Run <Operation> Dialog Box on page 69
- ➤ Manual Runner Service Testing Tab on page 71
- ➤ Specify Service Emulation Host Dialog Box on page 75

# Select Service from Systinet 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.
Relevant tasks	"Import a service" on page 61
See also	<ul> <li>"Import Service Dialog Box" on page 68</li> <li>"Select Service from UDDI Dialog Box" on page 64</li> </ul>

#### **Chapter 3 •** Web Services

The following elements are included:

UI Elements (A-Z)	Description
Refresh List	Reloads the list of the services on the Systinet server, in the <b>Available Services</b> pane.
Filter	Filters the list of services to show only those services containing the string specified in the <b>Service name</b> field.
Systinet Address	The name or IP address of the Systinet server.
	For Example: http://lab3.devlab.ad: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

Enables you to select and import services from a UDDI (Universal Description, Discovery, and Integration) server.

To access	In the Import Service dialog box, choose <b>Import WSDL from</b> : <b>UDDI</b> and click the Browse button.
Relevant tasks	"Import a service" on page 61
See also	"Select Service from Systinet Dialog Box" on page 63

The following elements are included:

UI Elements (A-Z)	Description
Refresh List	Reloads the list of the services from the UDDI, in the <b>Available Services</b> pane.
Filter	Filters the list of services to show only those services containing the string specified in the <b>Service name</b> field.
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

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

To access	<ul> <li>For a new service: In the New Component window (Components &gt; New &gt; New Component), select the Interaction node and click Browse. Click the Connection Settings button.</li> <li>For existing services: Select the Interactions tab, click Update From, and click Connection Settings.</li> <li>In Customization: Service Test Management link &gt; General Tab &gt; Edit Default Settings</li> </ul>
Important information	Only available for services imported through a URL and UDDI.
Relevant tasks	"Import a service" on page 61
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 133.

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
	<ul> <li>Use default credentials. authenticates with the Windows logon credentials of the user logged on to the machine</li> <li>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/alex_qc.</li> </ul>
Proxy	Access the WSDL using the following proxy server and credentials. Enables you to enter proxy details and credentials:
	<ul> <li>Proxy server. Name or IP address of proxy server.</li> <li>Port. Port through which to access the WSDL.</li> <li>Use default credentials. Connects to the proxy server with the Windows logon credentials of the user logged on to the machine.</li> </ul>
	➤ <b>User name, Password.</b> 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/alex_qc.
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	In the New Application Component window (Components > New > New Component), select the Interaction link. Click Browse.
Relevant tasks	"Import a service" on page 61.
See also	<ul> <li>"Connection Settings Dialog Box" on page 66</li> <li>"Select Service from Systinet Dialog Box" on page 63</li> <li>"Select Service from UDDI Dialog Box" on page 64</li> </ul>

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:
	<ul> <li>File. An Open File dialog box</li> <li>URL. A new browser window</li> <li>Systinet. Select Service from Systinet dialog box</li> <li>UDDI. Select Service from UDDI dialog box</li> </ul>
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 66.

UI Elements (A-Z)	Description
Import WSDL from	The WSDL source:
	➤ File. The file system
	➤ URL. A URL
	➤ Systinet. A Systinet registry
	➤ <b>UDDI</b> . A UDDI server
WSDL Location	Location of WSDL:
	➤ File: full path and file name
	➤ URL: Complete URL
	➤ Systinet: Systinet ID
	➤ UDDI: Server key

# 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   Web Service Call button.
Important information	You must have the Service Test add-in to use this utility.
Relevant tasks	"Test the service - optional" on page 62.

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 70.
Response Pane	The SOAP response. For details, see the "Run Operation - Response Pane" on page 71.

### **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 <b>Grid</b> tab.
Import XML	Allows you to import an XML structured file, for the purpose of loading values for the request.
Export XML	Allows 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.
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

UI Elements (A-Z)	Description
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	Enables you to export the SOAP response to an XML file.
SOAP Tab	SOAP response resulting from the request.

# Nanual Runner Service Testing Tab

This tab lets you manually run a Web Service test step.

You can verify the functionality of a specific operation using Spot testing. For details, see "Spot Testing Operations" on page 58.

To access	Perform the following steps:
	<ul> <li>Select a test set in the Test Lab module and click the Execution Grid tab.</li> </ul>
	➤ Select a test and choose Run > Run Manually.
	➤ Click the Service Testing tab.
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 62
See also	"Create an empty test - optional" on page 95

### **Chapter 3 •** Web Services

User interface elements are described below.

UI Elements (A-Z)	Description
4	<b>Previous Step.</b> Navigates to the previous test step.
	Next Step. Navigates to the next test step.
<i>€</i> •	<b>Set Pass Status.</b> Sets the test status to <b>Pass</b> . The drop down list provides the following options:
	<ul> <li>Pass Selected. Sets status to Pass for selected steps.</li> <li>Pass All. Sets status to Pass for all steps of the test.</li> </ul>
€ -	<b>Set Fail Status.</b> Sets the test status to <b>Fail</b> . The drop down list provides the following options:
	<ul> <li>Fail Selected. Sets status to Fail for selected steps.</li> <li>Fail All. Sets status to Fail for all steps of the test.</li> </ul>
2	Opens the New Defect dialog box for creating a defect for the current test and test set.
<b>€</b> •	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.
	Stop run. Terminates the manual test run.
Actual Result	The actual value returned. If this value matches the Expected Value, mark the status to <b>Pass</b> . Otherwise, set the status to the appropriate value.
	For a Web service step, use the <b>Send Request</b> button to check the response. See "Manual Runner - Service Testing Tab Request Pane" on page 73.

UI Elements (A-Z)	Description
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 73.
Response Pane	The SOAP response. For details, see the "Manual Runner - Service Testing Tab Response Pane" on page 74.
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 <b>Available Services and Operations</b> pane. In the <b>Grid</b> tab, provide argument values. If required, modify the XML and then send the request.
de Clear	Clears the values you assigned to the arguments in the <b>Grid</b> tab.
₺ Import XML	Enables you to import the request data from an XML file.

### **Chapter 3 •** Web Services

UI Elements (A-Z)	Description
Export XML	Enables you to export the request to an XML file.
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.
	<ul> <li>An argument with an ABC icon can be parameterized.</li> <li>A NIL icon indicates that argument can have a NIL value.</li> </ul>
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	Enables you to export the SOAP response to an XML file.
SOAP Tab	SOAP response resulting from the request.

# Specify Service Emulation Host Dialog Box

Enables you to select a host machine and port for your emulated service.

To access	Components > Emulation > Emulate Service.
Important information	The host machine must have the Tomcat server. This is installed automatically on the localhost with Service Test or the Service Test add-in.
Relevant tasks	"Create an emulated Web service - optional" on page 61
See also	For more information, see the Service Test User Guide.

The following elements are included:

GUI Element (A-Z)	Description
	<b>Browse.</b> Opens the Browse for Computer dialog box.
Host Name	The name or IP address of the machine to host the emulated service.
Port	The port number for the specified host machine  Default value: 8080  Tip: If you have a conflict with the 8080 port, edit the  Server.xml file in the apache-tomcat-5.5.17\conf folder.  Change the port from 8080 to another port. Create a new localhost server with the new port number.

## **Chapter 3 •** Web Services

# **Managing Components in Quality Center**

#### This chapter includes:

#### Concepts

➤ Managing Components on page 77

#### **Tasks**

➤ How to Manage Components on page 80

## **Concepts**



## Managing Components

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 Components tree, you can reorganize the components under different folders.

#### This section includes:

- ➤ "Component Details" on page 78
- ➤ "Locating Components" on page 78
- ➤ "Handling Rule Violations or Change Alerts" on page 78
- ➤ "Copying Components" on page 79
- ➤ "Attachments" on page 79
- ➤ "Restoring and Removing Components" on page 79
- ➤ "E-mailing Components" on page 80

## **Component Details**

In the **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 80.

## **Locating Components**

You can search for a particular entity in the 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 81.

## **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 81.

## **Copying Components**

You can copy an existing folder, component, or group and paste it to another location in the 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 additional information, see "Copy entities - optional" on page 81.

### **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 Quality Center 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 108. If you try to delete components that are used by one or more tests, Quality Center 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 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 79.

## **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 "Quality Center Direct Links" on page 26.

For additional information, see "Mail components- optional" on page 82.

## **Tasks**

## **P** 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" on page 80
- ➤ "Find folders and components optional" on page 81
- ➤ "Handle alerts optional" on page 81
- ➤ "Copy entities optional" on page 81
- ➤ "Remove and restore components optional" on page 82
- ➤ "Mail components- optional" on page 82

## Manage component details - optional

Open the Details tab and modify the component details as required.

For more information, see the "Details Tab" on page 29.

### 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 35.

### **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 28.

## **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 Component tree and paste the entity from the clipboard. To copy to another project, open it in another browser.

For details, see the "Application Components Module Window" on page 48.

### 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.

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 position 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 Quality Center User Guide*.

# 5

# Groups

This chapter includes:

#### Concepts

➤ Groups Overview on page 83

#### **Tasks**

➤ How to Manage Component Groups on page 85

#### Reference

➤ New Groups User Interface on page 87

## **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 Quality Center User Guide*.

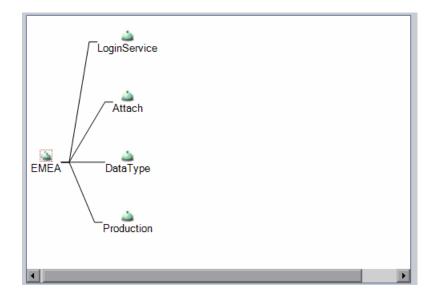
This section also includes:

- ➤ "Associating Components to Groups" on page 84
- ➤ "Group Coverage" on page 85

## **Associating Components to Groups**

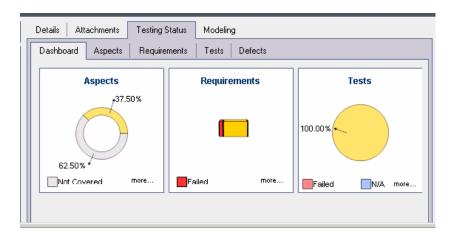
Groups help you organize your components into logical units. You might, for example, organize them into groups representing different physical deployments, business processes, or application types. After you create groups, you can associate components to the group, and perform tasks upon the entire group. 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 36.



## **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 86
- ➤ "Add a component to a group" on page 86
- ➤ "View and Modify Group Details optional" on page 86

- ➤ "Mail Group Information Optional" on page 86
- ➤ "Results and Coverage" on page 87

### **Create an application component group**

Create a group at the root level of the component or under a specific folder. Quality Center displays the groups in the Component tree hierarchy. For details, see "New Group Dialog Box" on page 87.

### 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.

For more details, see the "Modeling Tab" on page 36

## **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 31.

## **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 Quality Center, see the *Quality Center User Guide*.

### **Results and Coverage**

Click the **Testing Status** tab to view the complete coverage for aspects, requirements, or tests of the group's components.

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

## Reference

## New Groups User Interface

This section includes:

➤ New Group Dialog Box on page 87

## New Group Dialog Box

Enables you to create a new group.

To access	Use one of the following:
	<ul> <li>Select Components &gt; New &gt; New Group</li> <li>Select New &gt; New Group from the shortcut menu</li> </ul>
Relevant tasks	"Create an application component group" on page 86

The following elements are included:

GUI Element (A-Z)	Description
× Clear All	<b>Clear All.</b> Clears all the fields in the dialog box so that you can redefine the group.
<u>Q</u>	Details module. Enables access to the Details and Description tabs.
Cip .	Attachments module. Lets you attach a file to the group. For details, see the <i>Quality Center User Guide</i> .

## **Chapter 5 •** Groups

GUI Element (A-Z)	Description
Submit	<b>Submit.</b> Creates the group whose details you specified.
Close	<b>Close.</b> 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 you assigned to the group.

# **Generating Requirements and Tests**

This chapter includes:

#### Concepts

➤ Generating Requirements and Tests Overview on page 89

#### **Tasks**

➤ How to Generate Requirements and Tests on page 94

#### Reference

➤ Generate Requirements and Tests User Interface on page 95

## **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.

#### **Chapter 6 •** Generating Requirements and Tests

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 123

This section also includes:

- ➤ "Testing Aspects" on page 91
- ➤ "Empty Tests" on page 93
- ➤ "Actions after Creating a Test" on page 93

## **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.
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:
	<ul> <li>NET Framework. Tests that the service is fully interoperable with .NET Framework WSE 2 toolkit.</li> <li>Axis/Java Based Web Services. Tests that the service is fully interoperable with Axis 1.3 Web Services.</li> </ul>
Security Testing	Tests security. Contains the following sub-aspects:
	<ul> <li>SQL Injection Vulnerability. Checks if the component is vulnerable to SQL injections by injecting SQL statements and errors into relevant parameters.</li> <li>Cross-site Scripting (XSS). Attempts to hack the component by injecting code with relevant parameters that will disrupt its functionality.</li> </ul>

**Chapter 6 •** Generating Requirements and Tests

Aspect Name	Description
<b>Boundary Testing</b>	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:
	<ul> <li>Extreme Values. Provides invalid data types to the components and verifies they are not accepted.</li> <li>Null Values. Provides NULL parameters to the components to verify they are not accepted.</li> </ul>
Performance Testing	Contains the following sub-aspects:
	➤ <b>Stress Testing</b> . 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 145.

## **Empty Tests**

After you create a component in the **Application Components** module, you can create an empty test, edit it, and assign it to requirements at a later stage. For details on creating tests, see "Create Empty Test Dialog Box" on page 96.

## **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 Quality Center User's 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 <b>Test Lab</b> module. For details, refer to the <i>HP Quality Center User's Guide</i> .
Configure parameters for test instances	You can configure parameters for test instances in the <b>Test Lab</b> module. For details, refer to the <i>HP Quality Center User's Guide</i> .
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 123.

#### **Chapter 6 • Generating Requirements and Tests**

Action	Description
Manually add test coverage	You can manually add test coverage for a component. For details, see "Linking Components and Requirements" on page 123
Run the test	Run the test from Quality Center's <b>Test Lab</b> module. For details, see the <i>Quality Center User Guide</i> .

## **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 95
- ➤ "Run the wizard optional" on page 95
- ➤ "Review the generated tests" on page 95
- ➤ "Create an empty test optional" on page 95
- ➤ "Remove unwanted tests or requirements optional" on page 95

### 1 Prerequisites

Make sure you have at least one component in the 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 automatically create test scripts, select **Components > Generate Requirements/Tests** to open the wizard. For details, see the "Generate Requirements and Tests Wizard" on page 97.

### 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 Create an empty test - optional

If your component is not ready yet, you can create an empty test and modify it at a later point. Select **Components** > **Create Empty Test**.

## 5 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:

- ➤ Create Empty Test Dialog Box on page 96
- ➤ Generate Requirements and Tests Wizard on page 97

## Create Empty Test Dialog Box

This dialog enables you to create an empty test for a component.

To access	Components > Create Empty Test
Relevant tasks	"Create an empty test - optional" on page 95

User interface elements are described below:

UI Elements	Description
<u></u>	Opens the Select Test Folder Dialog Box.
Test Name	The name of the test as it will appear in the Tests tree and repository.
Test Folder	The folder under which to create the tests.

## **Select Test Folder Dialog Box**

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

UI Elements (A-Z)	Description
5	Refresh the Tests list.
7	Set or clear a filter for the Tests list.
<b>*</b>	Create a new folder.
<test folder="" list=""></test>	A list of the test folders in the tree hierarchy. Use the expand and collapse buttons to show or hide the folder contents.

# Generate Requirements and Tests Wizard

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

To access	Components > Generate Requirements/Tests
Relevant tasks	"Generate Requirements and Tests Wizard" on page 97
Wizard map	This wizard contains:  Select Entities Page > Select Aspects Page > Select Locations Page > Summary Page> Generate Page
See also	<ul> <li>"View the Requirements coverage - optional" on page 110</li> <li>"View the Tests coverage - optional" on page 111</li> </ul>

## Select Entities 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 97.
Wizard map	This wizard contains:
	Select Entities Page > <b>Select Aspects Page</b> > Select Locations Page > Summary Page> Generate Page

## **Chapter 6 •** Generating Requirements and Tests

User interface elements are described below:

UI Elements (A-Z)	Description
Generate	The entities to generate:
	<ul> <li>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.</li> <li>Tests only. Generates a set of tests for the aspects</li> </ul>
	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	The engine to use for generating a test.
Generator	<ul> <li>Manual Test. Creates manual tests with no design steps. This is available for all application component types.</li> <li>HP Service Test. Creates tests based on an installation of Service Test or the Service Test add-in</li> </ul>
	for Quality Center. See the Installation Guide for details (only available for Web Service type components).
	<custom generator="">. Creates tests based on a custom generator. For details, see "Custom Script Generators" on page 185.</custom>

# Select 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 97.
Wizard map	This wizard contains:  Generate Requirements and Tests Wizard > Select  Aspects Page > Select Locations Page > Summary Page> Generate Page
See also	<ul> <li>"Customizing Service Test Management" on page 133</li> <li>"Custom Script Generators" on page 185</li> </ul>

User interface elements are described below:

UI Elements (A-Z)	Description
	Clear aspect. Excludes the aspect.
<b>≥</b> ≱	<b>Select Aspect</b> . 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 <b>Expand</b> → and <b>Collapse</b> 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 97.
Wizard map	This wizard contains:
	Select Entities Page > Select Aspects Page > Select Locations > Summary Page > Generate Page

User interface elements are described below:

UI Elements (A-Z)	Description
<u></u>	<b>Browse</b> . Locate a parent requirement or test folder.
<pre><parent requirement=""></parent></pre>	Generate requirements as children of the selected requirement.
<tests folder=""></tests>	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 <b>Next</b> in this screen, the wizard begins generating the requirements and tests.
Wizard map	This wizard contains:
	Select Entities Page > Select 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 <b>Testing Aspects</b> 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.

## **Q** 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 97.
Wizard map	This wizard contains:
	Select Entities Page > Select Aspects Page > Select Locations Page > Summary Page > Generate Page

User interface elements are described below:

UI Elements (A-Z)	Description
Finish	Closes the wizard.

## **Chapter 6 •** Generating Requirements and Tests

UI Elements (A-Z)	Description
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
	➤ <b>Testing aspect</b> : See "Testing Aspects" on page 91.
	➤ Component: Component name
Generation Status Details	Provides the reasons for failed generation of requirements or tests.

# **Determining Test Coverage**

### This chapter includes:

### **Concepts**

- ➤ Coverage Overview on page 104
- ➤ Rule Violations on page 104
- ➤ Aspect Coverage on page 105
- ➤ Requirements Coverage on page 106
- ➤ Operations Coverage on page 107
- ➤ Tests Coverage on page 108
- ➤ Defects Coverage on page 109

#### **Tasks**

➤ How to Check the Testing Status on page 109

#### Reference

➤ Testing Status Tab on page 112

## **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 104
- ➤ "Aspect Coverage" on page 105
- ➤ "Requirements Coverage" on page 106
- ➤ "Operations Coverage" on page 107
- ➤ "Tests Coverage" on page 108
- ➤ "Defects Coverage" on page 109

## **A 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 111.

## **&** 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 want to see the coverage in meaningful terms.

The following table describes the aspect statuses.

Aspect Status	Description
All Failed	All the aspect's requirements are assigned the <b>Failed</b> status.
Some Failed	At least one of the aspect's requirements is assigned the <b>Failed</b> status.
All Passed	All the aspect's requirements are assigned the <b>Passed</b> status.
Some Passed	At least one of the aspect's requirements is assigned the <b>Passed</b> status. All other requirements have either the <b>No Run</b> , <b>Not Completed</b> , or <b>Not Covered</b> 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 <b>Not Covered</b> status.

## **& Requirements Coverage**

Requirement coverage is used to connect components to other Quality Center 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 89.

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 Quality Center coverage model and want to integrate application components into that model.

The following	table	describes	the rec	uirement	statuses.
1110 10110 111115	tubit	acocitoco	tire rec	1 0011 0111 0110	otatases.

Requirement Status	Description
Failed	At least one test associated with the requirement has failed.
N/A	The terms <b>Passed</b> and <b>Failed</b> do not apply to the requirement.
No Run	All tests associated with the requirement have a <b>No Run</b> status.
Not Completed	All tests associated with the requirement have the <b>Not Completed</b> 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 coverage for the components, per its operations. This information is dynamic and displays the current status in percentages. 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 script uses this operation.

#### **Chapter 7 • Determining Test Coverage**

The following table lists the possible statuses:

Status	Description
Failed	One or more of the tests associated with this operation has failed.
No Run	All tests associated with the operation have a <b>No Run</b> status.
Not Completed	All tests associated with the operation have the <b>Not Completed</b> status.
Not Covered	There are no tests associated with the operation.
Passed	All test associated with the operation have passed.

## **&** Tests Coverage

Tests describe the test objective and an implementation of the testing objective. They include the test operations and the expected outcome of each test. You can generate tests and link them to your components, or you can create a set of tests that are automatically linked to components through requirements (see Chapter 6, "Generating Requirements and Tests").

When linking directly to tests, you link tests to a component in the **Tests** section of the **Testing Status** tab. Alternatively, you can link components to tests in the **Test Plan** module.

**Note:** You can create 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 Quality Center Administrator's Guide*.

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.

# **& Defects Coverage**

The **Testing Status** section **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 *Quality Center User Guide*.

## **Tasks**

# **P** 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, tests, operations, and defects associated with your component.

This task includes the following steps:

- ➤ "View the Aspects coverage optional" on page 110
- ➤ "View the Requirements coverage optional" on page 110
- ➤ "View the Operations coverage optional" on page 110
- ➤ "View the Tests coverage optional" on page 111
- ➤ "View the Rule Violations optional" on page 111
- ➤ "View the Defects coverage optional" on page 112

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.
- **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 Quality Center* User Guide.

For user interface details, see the "Aspects Tab" on page 115.

#### 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 97 to create them. To add existing requirements to the coverage, click **Select Entities** to open the Requirements tree, 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 the **Application Components** module in the left pane.

For details about the Requirements tab, see "Requirements Tab" on page 116.

## **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 Quality Center* User Guide.
- **4** To return to your last view, click the **Application Components** module button in the left pane.

For UI details about the Operations tab, see the "Operations tab" on page 118.

#### **View the Tests coverage - optional**

- 1 Select a component, group, or folder and click the **Testing Status** > **Tests** tab.
- **2** To add more tests for a component, click **Select Entities**. Expand the tree in the **Test Plan** tab 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. Drag the test to the Test Coverage grid, or click the **Add to Coverage** button.
- **3** Click on a test to view the test in the **Test Plan** module. To return to your last view, click the **Application Components** module button.

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

For information about the **Tests** tab, see the "Tests Tab" on page 119.

### **View the Rule Violations - optional**

- **1** Select a component in the Application Components tree.
- **2** Select the **Testing Status** tab and click the **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 97.

For user interface details, see the "Rules Tab" on page 114.

#### View the Defects coverage - optional

- **1** Select a component or group in the tree hierarchy.
- **2** Select the **Testing Status** tab and click the **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 120.

For more information about the **Requirements**, **Test Plan**, and **Defects** modules, see the *HP Quality Center* User Guide.

## Reference

# Testing Status Tab

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

- ➤ "Testing Status Dashboard" on page 113
- ➤ "Rules Tab" on page 114
- ➤ "Aspects Tab" on page 115
- ➤ "Requirements Tab" on page 116
- "Operations tab" on page 118
- ➤ "Tests Tab" on page 119
- ➤ "Defects Tab" on page 120

# Testing Status Dashboard

The Coverage dashboard displays a summary of each of the coverage items: Aspects, Requirements, Operations, Tests, and Defects.

To access	Application Components module > <select an="" entity=""> &gt; Testing Status tab &gt; Dashboard</select>
Important information	Click on a graph to open it in a separate tab.
Relevant tasks	"How to Check the Testing Status" on page 109

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.
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.
Defects	A graphical representation of the coverage for the defects associated with the component.

# **Rules Tab**

This tab enables you to view and manage the rules violated by the selected component.

To access	Application Components module > <select an="" entity=""> &gt;Testing Status tab &gt; Rules</select>
Important information	See "Rule Violations" on page 104
Relevant tasks	"View the Rule Violations - optional" on page 111

UI Elements	Description
	<b>Select Columns.</b> Enables you to select the columns to display in the Rule Violations grid.
5	<b>Refresh.</b> Reloads the list of rule violations.
7	<b>Set Filter/Sort</b> . Enables you to set a filter for the rule violations or sort the changes by a criteria.
Resolve All	<b>Resolve All.</b> Opens the <b>Generate Requirements and Tests</b> wizard to resolve the rule violations. For details, see the "Generate Requirements and Tests Wizard" on page 97.
<rule grid="" violations=""></rule>	A list of all the rule violations for the selected component.
	<b>Default:</b> Rule Name and Rule Description.
	<b>Available Columns:</b> Alert ID, Alert is Active, and Component Name.

# **Aspects Tab**

Displays the coverage per testing aspect in both a graphical representation and a grid layout.

To access	Application Components module > <select an="" entity=""> Testing Status tab &gt; Aspects</select>
Important information	See "Aspect Coverage" on page 105
Relevant tasks	"View the Aspects coverage - optional" on page 110

GUI Element (A-Z)	Description
5	<b>Refresh.</b> Reloads the status of the testing aspects and refreshes the graph.
	<b>Collapse</b> . Shows the aspect only, but not its tests or requirements.
Ŧ	<b>Expand.</b> Shows the tests associated with the aspect.
<graph pane=""></graph>	A graphical representation of the requirement coverage
<grid pane=""></grid>	<ul> <li>A list of all aspects associated with the component.</li> <li>Expand aspect to see list of associated tests.</li> <li>Click test link to go to the test in the Test Lab module.</li> <li>Click the Application Components module to return to the most recent view of the component.</li> </ul>

# Requirements Tab

Displays the coverage per testing requirement in both a graphical representation and a grid layout.

To access	Application Components module > <select an="" entity=""> Testing Status tab &gt; Requirements</select>
Important information	See "Requirements Coverage" on page 106
Relevant tasks	"View the Requirements coverage - optional" on page 110

GUI Element (A-Z)	Description
♣ Select	Select. Opens the Requirements tree enabling you to add more requirements to your test. For details see "Requirements Tree" on page 117.
	Select Columns. Opens the Select Column dialog box enabling you to select the columns to display in the grid.
5	<b>Refresh.</b> 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.
*	<b>Collapse.</b> Collapses the graph window and show more of the grid.
*	<b>Expand.</b> Shows the graph window and displays less rows of the grid.

GUI Element (A-Z)	Description
<graph pane=""></graph>	A graphical representation of the requirement coverage.
<grid pane=""></grid>	A list of all requirements associated with the component.
	➤ Expand requirement to see list of associated tests
	<ul><li>Click requirement link to go to the requirement in the Requirements module.</li></ul>
	➤ Click test link to go to the test in the <b>Test Lab</b> module.
	<ul> <li>Click the Application Components module to return to the most recent view of the component.</li> </ul>

## **Requirements Tree**

Description
Add Coverage. Adds coverage for the selected requirement. If you select a parent requirement, it adds all sub-requirements.
<b>Refresh.</b> Refreshes the Requirements tree.
Set/Clear Filter. Sets or clears the filter.
<ul> <li>Set Filter / Sort. Opens the Set Filter/Sort dialog box allowing you to filter out and sort the requirements by any field.</li> <li>Clear Filter / Sort. Clears all filters and sorting settings and displays all requirements.</li> </ul>
Closes the Requirements tree pane.  To reopen this view, click the <b>Select</b> button in the Requirements tab.
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.

# **Q** Operations tab

Displays the coverage per operation in both a graphical representation and a grid layout.

To access	Application Components module > <select an="" entity=""> Testing Status tab &gt; Operations</select>
Important information	See "Operations Coverage" on page 107
Relevant tasks	"View the Operations coverage - optional" on page 110

GUI Element (A-Z)	Description
5	<b>Refresh.</b> Reloads the status of the operations and refreshes the graph.
⊡	<b>Collapse.</b> Shows the operation name only, but not its tests. If there are multiple ports, collapsing the node shows the port name, while hiding the operations.
H	<b>Expand.</b> Shows the operations and their testing status.
<graph pane=""></graph>	A graphical representation of the Operations coverage.
<grid pane=""></grid>	<ul> <li>A list of all operations associated with the component.</li> <li>Expand operation to see list of associated tests.</li> <li>Click test link to go to the test in the Test Lab module.</li> <li>Click the Application Components module to return to the most recent view of the component.</li> </ul>

# **1** Tests Tab

Displays the status per test in both a graphical representation and a grid layout.

To access	Application Components module > <select an="" entity=""> Testing Status tab &gt; Tests</select>
Important information	See "Tests Coverage" on page 108
Relevant tasks	"View the Tests coverage - optional" on page 111

GUI Element (A-Z)	Description
↑ Select	Select. Opens the Tests tree enabling you to associate more tests with your component. For details see "Tests Tree" on page 120.
	<b>Select Columns.</b> Opens the Select Column dialog box enabling you to select which columns to show in the grid.
5	<b>Refresh.</b> Reloads the status of the test and refreshes the graph.
×	<b>Remove.</b> Unlinks the test from the requirement through which it is associated with the component.
*	<b>Collapse.</b> Collapses the graph window and shows more of the grid.
*	<b>Expand.</b> shows the graph window and displays less rows of the grid
<graph pane=""></graph>	A graphical representation of the test coverage.
<grid pane=""></grid>	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 Components to return to the most recent view of the component.

### **Tests Tree**

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

GUI Element (A-Z)	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.
5	<b>Refresh.</b> Refreshes the Tests tree.
▼ •	<ul> <li>Set/Clear Filter. Sets or clears the filter.</li> <li>Set Filter / Sort. Opens the Set Filter/Sort dialog box allowing you to filter out and sort the tests.</li> <li>Clear Filter / Sort. clears all filters and sorting settings and displays all tests.</li> </ul>
×	Closes the Tests tree pane.  To reopen this view, click the <b>Select</b> button in the Tests tab.
<requirements tree=""></requirements>	A tree view of all of the tests by folder and group <b>Tip:</b> If a filter is active, there could be hidden tests. To see all tests, click <b>Clear Filter</b> .

# **Defects Tab**

Displays the status of the defects in both a graphical representation and a grid layout.

To access	Application Components module > <select an="" entity=""> Testing Status tab &gt; Defects</select>
Important information	See "Defects Coverage" on page 109
Relevant tasks	"View the Defects coverage - optional" on page 112

GUI Element (A-Z)	Description
	Select Columns. Opens the Select Column dialog box enabling you to select which columns to show in the grid.
5	<b>Refresh.</b> Reloads the status of the defects and refresh the graph.
*	<b>Collapse.</b> Collapses the graph window and show more of the grid.
*	<b>Expand.</b> Shows the graph window and displays less rows of the grid.
<graph pane=""></graph>	A graphical representation of the defects coverage.
<grid pane=""></grid>	A list of all defects associated with the component and their status.
	Click <b>Defect ID</b> link to open the defect in the <b>Defects</b> module.
	<ul> <li>Click the Application Components module to return to the most recent view of the defects coverage.</li> </ul>
	➤ Drag a column separator to adjust its width.

**Chapter 7 •** Determining Test Coverage

# **Linking Components and Requirements**

This chapter includes:

#### Concepts

➤ Linking Components Overview on page 123

#### **Tasks**

➤ How to Link Components to Requirements and Tests on page 124

#### Reference

➤ Linking Component User Interface on page 125

# **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 Quality Center 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 components linked to each requirements. The **Components** tab displays a list of those components, with other basic information.

#### **Chapter 8 • Linking Components and Requirements**

Using the **Test Lab** module's **Component** tab (available only after you install the Service Test Management extension), you can set and view the 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 Quality Center, 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 124
- "Link a component to a test optional" on page 124
- ➤ "Results" on page 125

### 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 127.

### Link a component to a test - optional

Use the **Application Components** tab in the **Test Plan** module to link a component to a test. For details, see "Application Components Tab - Test Plan Module" on page 130.

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 126.

You can also link a component to a test through the **Testing Status** tab. For details, see "Tests Coverage" on page 108.

#### 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 125.

## Reference

# Linking Component User Interface

This section includes (in alphabetical order):

- ➤ Link <test-name> to Requirement Dialog Box on page 126
- ➤ Application Components Tab Requirements Module on page 127
- ➤ Application Components Tab Test Plan Module on page 130

# Link <test-name> to Requirement Dialog Box

This dialog box enables you to select a requirement to associate with your test.

To access	Open the <b>Test Plan</b> module, and click the <b>Application Components</b> tab.
	Click Select and choose a entity from the Application Components tree.  Click the button.
Relevant tasks	"Link a component to a test - optional" on page 124
See also	"General Tab" on page 141 (Customization)

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

UI Elements (A-Z)	Description
	<b>Select Columns.</b> Opens the Select Column dialog box enabling you to select the columns to display in the grid.
5	<b>Refresh.</b> Reloads the components in the grid.

UI Elements (A-Z)	Description
<linked Requirements grid&gt;</linked 	A list of the requirements associated with the test.  To modify the columns, click the <b>Select Columns</b> button.
Link <test-name> to</test-name>	Sets a brokerage requirement option for linking to the test.
	➤ <b>Default Requirement.</b> Creates a link between the default requirement of the component and the test, if it does not already exist. The linked requirements grid is not available for this option.
	➤ Linked Requirement. Enables you to specify a requirement to link to the test. Select a requirement from the Linked Requirements grid. 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 110.

# Application Components Tab - Requirements Module

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

To access	<ul> <li>Open the Requirements module.</li> <li>Select View &gt; Requirement Details.</li> <li>Expand the tree and select a requirement.</li> <li>In the right pane, scroll right to Application Components tab.</li> </ul>
Relevant tasks	"How to Link Components to Requirements and Tests" on page 124
See also	"How to Check the Testing Status" on page 109

## **Chapter 8 • Linking Components and Requirements**

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

UI Elements (A-Z)	Description
- Select	Opens the <b>Application Components</b> tree in the right pane to link a component to the requirement. For details, see the "Components Tree (Requirements Grid)" on page 129.
×	<b>Delete.</b> Unlinks the component and requirement.
	<b>Select Columns.</b> Opens the Select Column dialog box enabling you to select the columns to display in the grid.
5	Refresh. Reloads the components in the grid.
<linked components="" grid=""></linked>	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.
	➤ <b>Assigned to QA.</b> 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.

## **Components Tree (Requirements Grid)**

UI Elements (A-Z)	Description
<b>4</b>	Add Component. Adds the selected entity to the Linked Components grid.
5	Refresh. Refreshes the Components tree.
7	<b>Set/Clear Filter.</b> Opens a dropdown list with the following options:
	<ul> <li>Set Filter / Sort. Opens the Set Filter/Sort dialog box allowing you to filter and sort the components</li> <li>Clear Filter / Sort. Clears all filters and sorting settings and displays all components and groups.</li> </ul>
A	<b>Find.</b> Searches for a component whose name matches the text in the search box.
×	Exit. Closes the Application Components tree pane To reopen this view, click the Select Select button.
<component tree=""></component>	A tree view of all of the component by folder and group. <b>Tip:</b> If a filter is active, there could be hidden components. To see all components, click <b>Clear Filter</b> .

# 

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

To access	Open the <b>Test Plan</b> Module.
	Select View > Test Plan Tree.
	Expand tree and select a test in the left pane.
	In the right pane, scroll right to the <b>Application Components</b> tab.
Relevant tasks	"How to Link Components to Requirements and Tests" on page 124
See also	"How to Check the Testing Status" on page 109

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

UI Elements (A-Z)	Description
- Select	Opens the <b>Application Components</b> tree in the right pane to enable linking a component to the test. For information about the Components tree, see "Component Tree (Tests Grid)" on page 131.
×	<b>Delete.</b> Unlinks the test from all requirements that associate it with the component.
	<b>Select Columns.</b> Opens the Select Column dialog box enabling you to select the columns to display in the grid.

UI Elements (A-Z)	Description
5	<b>Refresh.</b> Reloads the list in the grid.
<li><li>linked components grid&gt;</li></li>	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.
	➤ <b>Component name.</b> 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).
	➤ <b>Assigned to QA.</b> 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.

# **Component Tree (Tests Grid)**

UI Elements (A-Z)	Description
<b>4</b>	Add Component. Adds the selected component to the Linked Components grid.
5	<b>Refresh.</b> Refreshes the <b>Application Components</b> tree in the right pane.
7	<b>Set/Clear Filter.</b> Opens a dropdown 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.

## **Chapter 8 •** Linking Components and Requirements

UI Elements (A-Z)	Description
M	<b>Find.</b> Searches for a component whose name matches the text in the search box.
×	<b>Exit.</b> Closes the <b>Application Components</b> tree pane.  To reopen this view, click the <b>Select</b> button.
<application components="" tree=""></application>	A tree view of all of the application components by folder and group (only visible when you click <b>Select</b> ).
	<b>Tip:</b> If a filter is active, there could be hidden components. To see all components, click <b>Clear Filter</b> .

# **Customizing Service Test Management**

#### This chapter includes:

#### **Concepts**

➤ Customization Overview on page 133

#### **Tasks**

➤ How to Customize a Project on page 138

#### Reference

➤ Project Customization User Interface on page 140

# **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, and user-defined testing aspects.

You can also customize component and group entities in the same way as you can customize other entities in your Quality Center projects. 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.

#### **Chapter 9 • Customizing Service Test Management**

In addition, you can customize access to the **Application Components** module for each user group, and customize the system fields and add new user-defined fields.

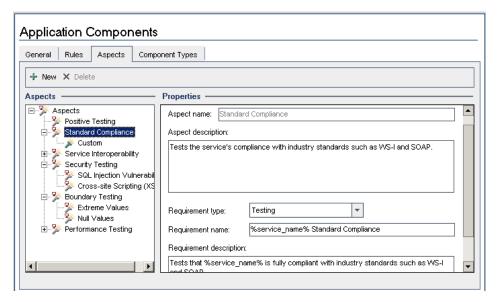
**Note:** This chapter describes the additional customization options available with Service Test Management. For details on Project Customization, refer to the *HP Quality Center Administrator's Guide*.

#### This section also includes:

- ➤ "Testing Aspects" on page 135
- ➤ "Rules" on page 136
- ➤ "Service Test Management Permissions" on page 137
- ➤ "Application Components Module Access" on page 137
- ➤ "Custom Project Entities" on page 137

### **Testing Aspects**

You can customize the template for the system and user-defined testing aspects displayed in the Requirement and Test Generation wizard. Using the Aspects tab, you can add new aspects, modify the properties of system and user-defined aspects, and delete user-defined aspects.



In the Aspects tree hierarchy, the  $\nearrow$  icon indicates a system-defined aspect. The  $\nearrow$  icon indicates a user-defined aspect.

For information on defining aspects, see "Customize the aspects - optional" on page 139.

#### 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 Quality Center filters. For details, see the *HP Quality Center User's Guide*.
- ➤ **Aspects.** The testing aspects that meet the filter criteria, 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 139.

### **Service Test Management Permissions**

You can customize the permission settings for user groups relating to their access to components, linkage to requirements, and ability to make changes. Built-in user group permissions are read-only, but you can create your own groups with custom permissions. For details about setting permissions, see "Customize group permissions - optional" on page 140.

For more information about user group permissions, refer to the *HP Quality Center Administrator's Guide*.

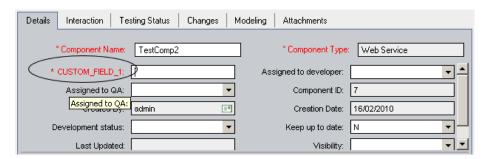
## **Application Components Module Access**

After you enable the HP Service Test Management extension for a project, you can control access to the **Application Components** module for each user group. You can enable access to all Quality Center modules including the **Application Components** module, or to all Quality Center modules excluding the **Application Components** module.

For details about customizing module access for user groups, refer to the *HP Quality Center Administrator's Guide*.

## **Custom Project Entities**

You can customize the system fields and create new user-defined fields for your project. You can create new fields for your component and indicate whether or not a field is mandatory. If you define a field as required, the **Details** tab shows it as mandatory.



For task details about customizing project entities, see "Customize project entities - optional" on page 140 and the *HP Quality Center Administrator's Guide*. For more information on Quality Center entities and fields, refer to the *HP Quality Center Database Reference*.

## **Tasks**



# How to Customize a Project

This task describes how to customize your Quality Center 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 138
- ➤ "Customize the General setting optional" on page 139
- ➤ "Customize the component rules optional" on page 139
- ➤ "Customize the aspects optional" on page 139
- ➤ "Define new component types optional" on page 139
- ➤ "Customize group permissions optional" on page 140
- ➤ "Customize application component module access optional" on page 140
- ➤ "Customize project entities optional" on page 140

### **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 Quality Center window.

#### **Customize the General setting - optional**

Define the parsing order for toolkits, and the default servers that host the WSDL, UDDI registry, or Systinet registry.

Click the **Application Components** link. Select the **General** tab. For user interface details, see the "General Tab" on page 141.

#### **Customize the component rules - optional**

Define a rule that lists the testing aspects to which your components must comply, for all components in the project. If the component does not meet the requirements of the rule, it issues a rule violation. For details, see "Rule Violations" on page 104.

Click the **Application Components** link. Select the **Rules** tab. For user interface details, see the "Rules Tab" on page 143.

#### **Customize the aspects - optional**

Manage the built-in and custom testing aspects. You can edit the properties of the built-in aspects or define new ones.

Click the **Application Components** link. Select the **Aspects** tab. For user interface details, see the "Aspects Tab" on page 145.

### **Define new component types - optional**

Create new component types in addition to the built-in ones: General, Web Services, and JMS.

Click the **Application Components** link. Select the **Component Types** tab. For user interface details, see the "Component Types Tab" on page 147.

#### **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 Pane" on page 151.

Click the **Groups** link. Select a user-defined group (not a built-in group) and click the **Change** button. Scroll to the right and select the **Application Components** tab. Modify the permissions as required.

#### **Customize application component module access - optional**

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

Click the **Module Access** link. Select a user-defined group (not a built-in group) and double-click the **Application Components** column to enable or disable access to the module.

#### **Customize project entities - optional**

Define new fields for the application component, application group, and application changes entities. Indicate if providing values is mandatory or optional. For details, see the "Project Entities Pane" on page 148.

## Reference

# Project Customization User Interface

This section includes (in alphabetical order):

- ➤ General Tab on page 141
- ➤ Rules Tab on page 143
- ➤ Aspects Tab on page 145
- ➤ Component Types Tab on page 147
- ➤ Project Entities Pane on page 148

- ➤ Groups Pane on page 151
- ➤ Module Access Pane on page 154

# **Q** General Tab

This tab enables you to customize the settings for 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.

To access	Tools > Customize and click the Application Components link. Select the General tab.
Important information	To resume work in the Quality Center modules, click the Beturn button in the top right corner.
Relevant tasks	"Customize the General setting - optional" on page 139

User interface elements are described below:

UI Elements	Description
Connection Settings	<ul> <li><description>. Text indicating the connection settings for WSDL imports.</description></li> <li>Edit Default Settings . Opens the Connection Setting dialog box for setting the default connection settings. For details, see "Connection Settings Dialog Box" on page 66.</li> </ul>
Default Systinet Server	Server Name. The default path of the Systinet server when a user opens the Select Service from Systinet Dialog Box.  Note: Systinet versions 2-3.x are supported.
Default UDDI Server	<ul> <li>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.</my_domain></my_server></li> <li>Version. The UDDI version: 2 or 3.</li> </ul>

UI Elements	Description
Auto-Detect Toolkit Order	<b>Parsing Order.</b> The order in which to try toolkit parsers when importing a WSDL in Auto-detect mode.
Default Requirements	<ul> <li>Location. The default requirement to use when linking a component to a test. The Browse button opens the Select Requirement dialog box.</li> <li>Requirement Type. One of the standard requirement types: Functional, Testing, or Undefined.</li> </ul>
Fuzzy Import	<ul> <li>Use fuzzy logic on import. For .NET services, use a looser set of criteria for parsing the components.</li> <li>Number of Attempts. The number of attempts to try parsing the WSDL before aborting the import.</li> </ul>

## **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
5	Refresh. Refreshes the selected requirement.
7	<b>Set/Clear Filter.</b> Opens a dropdown list with the following options:
	<ul> <li>Set Filter / Sort. Opens the Set Filter/Sort dialog box allowing you to filter and sort the requirements.</li> <li>Refresh All. Refreshes all of the requirements in the</li> </ul>
	<ul><li>tree.</li><li>Clear Filter / Sort. Clears all filters and sorting settings and displays all requirements.</li></ul>
A	<b>Find.</b> Finds the requirement containing the specified text.
<requirements tree=""></requirements>	Tree hierarchy of all the requirements.

# **Rules Tab**

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

To access	<b>Tools &gt; Customize</b> and click the <b>Application Components</b> link. Select the <b>Rules</b> tab.
Important information	To resume work in the Quality Center modules, click the button in the top right corner.
Relevant tasks	"Customize the component rules - optional" on page 139
See also	"Alerts Dialog Box" on page 28

User interface elements are described below:

UI Elements	Description
+ New	Opens the Create New Rule dialog box.
× Delete	Removes the selected rules from the Rule list.

## **Chapter 9 •** Customizing Service Test Management

UI Elements	Description
Rules	A list of the rules defined for the current project.
Properties	The properties of the rule:
	➤ <b>Rule</b> . 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 include all components in the project, define a filter to display all components with a Component ID > 0. To set a condition, click the Edit button to open the Filter Dialog Box.
	➤ <b>Aspects.</b> The testing aspects that must be covered for the component. To select the aspects, click the  Edit button to open the Select Aspect dialog box.
	➤ <b>Description</b> . Text describing the rule. To allow Service Test Management to generate a description based on the selected aspects and conditions, select <b>Use</b> Automatic Text.

## **Filter Dialog Box**

User interface elements are described below:

UI Element s(A-Z)	Description
7	Clear Filter. Clears all conditions to which the rule should apply.
I <sub>h</sub>	<b>Copy filter.</b> 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.
À	Paste filter. Uses a filter from the clipboard.

UI Element s(A-Z)	Description
Field Name	The field names to which you can apply a conditions. The field names correspond to the component details in the Project Entities list.
Filer Condition	The condition that must be met. You can use asterisks (*) as wild cards.

# Aspects Tab

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

To access	Tools > Customize and click the Application Components link. Select the Aspects tab.
Important information	To resume work in the Quality Center modules, click the Return button in the top right corner.
Relevant tasks	"Customize the aspects - optional" on page 139
See also	"Alerts Dialog Box" on page 28

User interface elements are described below:

UI Elements	Description
+ New	Opens the Create New Aspect dialog box.
× Delete	Removes the selected aspect from the list. You can only delete user-defined aspects—not built-in ones.

**Chapter 9 •** Customizing Service Test Management

UI Elements	Description
Aspects	A list of the aspects that are available for the current project.
Properties	The properties of the rule:
	➤ <b>Aspect Name.</b> The aspect name as it will appear in the tree. For built-in aspects this field is read-only.
	➤ <b>Aspect Description</b> . Text describing the goal of the aspect.
	➤ Requirement type. The requirement type: Undefined,
	Business, Group, Functional, or Testing. If you plan to
	link tests to this requirement, for example, by generating requirements and tests at the same time,
	use the <b>Covering</b> type. For details, see the <i>Quality</i>
	Center User Guide.
	➤ Requirement name. A name for the requirement. To include the component name, use the variable %service_name%.
	➤ <b>Description</b> . Text describing the requirement. To include the component name, use the variable
	%service_name%.

# **Q** Component Types Tab

This pane provides information for 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.

To access	<b>Tools</b> > <b>Customize</b> and click the <b>Application Components</b> link. Click the <b>Component Types</b> tab.
Relevant tasks	"Define new component types - optional" on page 139

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 <b>Name</b> field.
× Delete	Removes the selected component from the list (except for <b>General</b> and <b>Web Services</b> types).
	<b>Note</b> : You cannot remove a type if a component of that type exists in the repository.
Replace	Opens the Choose Image for Application Component Type dialog box, enabling you to choose an icon to represent the type.
Component types	A list of the built-in and custom component types. The built-in types are General, Web Service, and JMS.

**Chapter 9 • Customizing Service Test Management** 

UI Elements (A-Z)	Description
Details	The component details:
	➤ Name. The name of the component as it will appear in Quality Center.
	➤ <b>Assigned Icon</b> . The icon representing the component in the Application Components tree. For details, see "Application Components Module Icons" on page 47.
	➤ Replace opens a dialog box for changing the icon.
	➤ Support file extensions. The file extensions supported as a contract for components of this type, for example, WSDL
	➤ Contract parser. The COM object of the parser GUID used for parsing a component definition file, such as the WSDL.
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.
	➤ <b>Required for Type</b> . 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 43.

# **Project Entities Pane**

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

> System fields. These are Quality Center 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 Quality Center entities and fields, refer to the *HP Quality Center Database Reference*.

To access	Tools > Customize and click the Project Entities link. Expand the Application Component, Application Component Group, or Application Component Change entities.
Important information	You can only create and delete <b>User</b> fields, but not <b>System</b> fields.
Relevant tasks	"Customize project entities - optional" on page 140
See also	"Application Component Details Pane" on page 39

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

UI Elements (A-Z)	Description
New Field	Opens a dropdown list allowing you select to a <b>New Field</b> or a <b>New Memo Field</b> .
Remove Field	Removes the selected User field.
Allow Multiple Values	For user-defined fields, this option allows the user to select more than one value in any field that is associated with a predefined lookup list. (Available only when the <b>Lookup List</b> type is selected.)
Field Label	The name of the field to display in the Project Entities tree and the component's Details tab.
Field Length	The field size. (Available for <b>String</b> type only.)
	Note: The maximum field length is 255 characters.
Field Name	The field name used in the Quality Center database table.

**Chapter 9 •** Customizing Service Test Management

UI Elements (A-Z)	Description
Field Type	<ul> <li>The type of data valid for the field:</li> <li>Number. Enables integer entry only.</li> <li>String. Enables the entry of any character string.</li> <li>Date. Enables the selection of a date.</li> <li>Lookup List. Displays the Lookup List area and enables the selection from a drop-down list.</li> <li>User List. Enables the selection of a user name from your Quality Center users list.</li> <li>Memo. Enables the entry of blocks of data. Note that you can add up to 5 memo fields to each Quality Center entity.</li> </ul>
Goto List	Displays a predefined list. (Available only when the <b>Lookup List</b> type is selected.) To open a predefined list, select a list from the Lookup List box and click the <b>Goto List</b> button.
History	Preserves a log of values entered in the selected field.
Lookup List	A list of predefined lists. (Available only when the <b>Lookup List</b> 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 <b>Goto List</b> button.
Masked	The input data mask for the field. (Available only when the <b>String</b> type is selected.)
New List	Creates a new list. (Available only when the <b>Lookup List</b> type is selected.) To associate a field with a new list, click the <b>New List</b> button. The Project Lists dialog box opens.
Required	Indicates that a user must enter a value for the field.  Note: If you set a field as required in a project that already contains data, users do not need to enter a value for the field when they modify an existing record if the field is already empty.

UI Elements (A-Z)	Description
Searchable	Indicates a searchable field. (Available only when the <b>Text Search</b> option is enabled in the <b>DB Servers</b> tab.
Verify List	Limits the user to select a value only from the items that are listed in the list box. (Available when <b>Lookup List</b> or <b>User List</b> is selected.)

# **Q** Groups Pane

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

To access	Tools > Customize and click the Groups link. Select a group.  To view only: Click View in the Permissions section.  To modify: Click Change. This is only available for user-defined groups.  Select the Application Components tab (you may need to scroll to the right).
Important information	To resume work in the Quality Center modules, click the Button in the top right corner.
Relevant tasks	"Customize group permissions - optional" on page 140
See also	"Customize application component module access - optional" on page 140

### **Chapter 9 •** Customizing Service Test Management

The **Application Components** tab displays the following tasks that are available in the **Application Components** module:

Task	Description
Add Component	User group can add components to the <b>Application Components</b> module.
Modify Component	User group can modify components in the Application Components module. Note that this task enables you to specify the fields that the selected user group can modify.  Tip: Expand this task to set permissions for specific fields within the tab.
Delete Component	User group can delete components from the Application Components module.
Link Component to Requirements	User group can link components to requirements in the <b>Application Components</b> module.
Update Component/Group	User group can update components or groups. Note that selecting this task automatically enables the <b>Modify Component</b> task.
Add Component Folder	User group can add folders to the <b>Application Components</b> tree.
Modify Component Folder	User group can modify folders in the <b>Application Components</b> tree. This task enables you to specify the fields that the selected user group can modify. <b>Tip:</b> Expand this task to set permissions for specific fields within the tab.
Delete Component Folder	User group can delete folders from the <b>Application Components</b> tree.
Add Group	User group can add groups to the <b>Application Components</b> module.

**Chapter 9 •** Customizing Service Test Management

Task	Description
Modify Group	User group can modify groups in the <b>Application Components</b> module. This task enables you to specify the fields that the selected user group can modify.
	<b>Tip:</b> Expand this task to set permissions for specific fields within the tab.
Delete Group	User group can delete groups from the <b>Application Components</b> module.
Link Group to Requirements	User group can link a group to a specific requirement.
<b>Emulate Component</b>	User group can emulate Web Service components from Quality Center.
Add Change	User group can add changes to the <b>Application Components</b> module.
Modify Change	User group can modify changes in the <b>Application Components</b> module. This task lets you specify the fields that the selected user group can modify.
	<b>Tip:</b> Expand this task to set permissions for specific fields within the tab.
Delete Change	User group can delete changes from the <b>Application Components</b> module.
Customize Application Components	User group can specify settings for the Application Components module, using the Application Components link in Project Customization.
Link Component to Component	User group can link one component to another.
Link Component to Group	User group can link a specific component to a group.

### Module Access Pane

This pane lists the module access per group. You can control access for the following modules: **Defects**, **Test Plan**, **Test Lab**, **Requirements**, **Business Components**, **Management**, and the **Application Components** modules.

For detailed information on module access, refer to the *HP Quality Center Database Reference*.

To access	Tools > Customize and click the Module Access link.
Relevant tasks	"Customize application component module access - optional" on page 140
See also	"Customize group permissions - optional" on page 140

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

UI Elements (A-Z)	Description
Enable All	Enables the selected group to access any of the Quality Center modules.
Disable All	Disables the selected group from accessing any of the Quality Center modules.
<module access="" grid=""></module>	A table indicating each group's permission per module.

# 10

# **Change Analysis**

This chapter includes:

### **Concepts**

➤ Change Analysis Overview on page 155

#### **Tasks**

➤ How to Perform Change Analysis on page 158

#### Reference

➤ Changes Tab on page 161

# 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 Quality Center. 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.

#### **Chapter 10 •** Change Analysis

į

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 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 114.



### Reporting Changes Manually

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

Using the Change Details dialog box, you must indicate the following items:

- **Subject**. A text box with a meaningful name for the change.
- **Affected part.** The name of the affected application component part or application group.
- ➤ Change Status. In Test, New, Obsolete, or Modified
- ➤ **Risk**. High, Medium, or Low
- ➤ Affected part type. Component, Field, Interface, or Operation

### ➤ Change type. Added, Updated, or Removed

You can also add descriptions and comments, and assign the change to developers or QA personnel. For details, see "New Change Dialog Box" on page 163.



### 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 Quality Center 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 57.

**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 160.

**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 158
- ➤ "Update a Component optional" on page 159
- "Review the changes" on page 160
- ➤ "Evaluate the tests" on page 160
- ➤ "Create regression tests" on page 160
- ➤ "Run the test sets" on page 160
- ➤ "Check the test status" on page 160
- ➤ "Update the change status" on page 161

### 1 Define a New Change - optional

To manually define a change in the Changes list:

**a** In the Application Components tree, select the component you want to update.



- **b** Choose the **Changes** tab and click the New Change button.
- **c** Fill in the following mandatory fields:
  - **Subject**. A text box with a meaningful name for the change.
  - ➤ Affected part. The name of the affected application component part.
  - ➤ Change Status. In Test, New, Obsolete, or Modified
  - ➤ **Risk.** High, Medium, or Low
  - ➤ **Affected part type.** Component, Field, Interface, or Operation
  - ➤ Change type. Added, Updated, or Removed

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

### 2 Update a Component - optional

To instruct Service Test Management to automatically generate a list of changes, update the component contract stored in Quality Center with the latest contract file. Service Test Management flags the changed component in the Components tree.

- **a** In the 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 Alerts icon to open the Alerts dialog box and view a list of the alerts. Click **Show** to open 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 165.

#### 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 111.

### 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 167

#### 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 Quality Center graphs, reports, and documents. For details on running test sets and analyzing test results, refer to the *HP Quality Center User's Guide*.

#### 7 Check the test status

Review the coverage status of the parent requirement created for the changes set in the Testing Status tab, and verify the changes. For details, see ""Requirements Coverage" on page 106.

### 8 Update the change status

Verify the behavior of the changed component and manually update the change status in the **Changes** tab.

- **a** Select an updated component in the Components tree, and click the **Changes** tab.
- **b** In the grid, select the change whose status you want to update.
- **c** Double-click in the **Change Status** column, 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 165.

### Reference

# **Q** Changes Tab

This section includes (in alphabetical order):

- ➤ Changes on page 162
- ➤ New Change Dialog Box on page 163
- ➤ Application Change Details Dialog Box on page 165
- ➤ Create Change Impact Test Set Wizard on page 167

# **Q** Changes

This pane enables you to view and manage the changes made to the application component.

To access	Application Components module, <select_entity> Changes tab.</select_entity>
Relevant tasks	"Review the changes" on page 160
Important Information	For folders and groups, the <b>Changes</b> tab is read-only.

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

UI Elements	Description
Δ.	New Change. Opens the New Application Change dialog box to create a new change upon your component. See the "New Change Dialog Box" on page 163.
<u>L</u>	Change Details. Opens the Application Change Details for viewing and editing. See the "Application Change Details Dialog Box" on page 165
×	<b>Delete.</b> Removes a change from the Changes grid.
	<b>Select Columns.</b> 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 165.
5	<b>Refresh.</b> Reloads the list of changes.
7	<b>Set Filter/Sort</b> . Enables you to set a filter for the changes or sort the changes by a criteria.

UI Elements	Description
reate Test Set	<b>Create Test Set.</b> Opens the Create Change Impact Test Set wizard. For details, see "Create Change Impact Test Set Wizard" on page 167.
<changes grid=""></changes>	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.
	<b>Default:</b> Subject, Affected Part Type, Affected Part, Change Type, Risk, Change Status. Creation Date, Assigned To, Release, and Cycle.

# New Change Dialog Box

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

To access	Changes tab > New Change △ button.
Relevant tasks	"Review the changes" on page 160
See also	"Changes" on page 162.

User interface elements are described below:

UI Elements (A-Z)	Description
× Clear	Clears all the fields in the dialog box.
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, Interface, Operation, or Field.
Assigned to	The user name of the person who is responsible for the change.

### **Chapter 10 •** Change Analysis

UI Elements (A-Z)	Description
Change status	The current status of the change. By default, the change status is <b>New</b> . 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.
	➤ <b>Obsolete.</b> 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.
	➤ <b>Updated</b> . 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.
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.
	<b>Default:</b> 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 <b>Part Type</b> is <b>Operation</b> and <b>Change Type</b> is <b>Updated</b> ), the change description contains details of the argument that was added or removed from the operation.

UI Elements (A-Z)	Description
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:
	➤ <b>High.</b> 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.

# Name Application Change Details Dialog Box

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

To access	Changes tab > Change Details 🏠 button.
Relevant tasks	"Review the changes" on page 160
See also	"Changes" on page 162.

User interface elements are described below:

UI Elements(A-Z)	Description
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.

### **Chapter 10 •** Change Analysis

UI Elements(A-Z)	Description
Change status	The current status of the change. By default, the change status is <b>New</b> . Update the status throughout the change analysis process as follows:
	<ul><li>In test. The change is being tested.</li><li>New. There is a change in the component.</li></ul>
	<ul> <li>Obsolete. The change does not affect the component.</li> <li>Verified. The change was tested and did not cause a regression.</li> </ul>
	<b>Tip:</b> To eliminate the Alert icon next to the component, change the status of all changes to <b>Verified</b> or <b>Obsolete</b> .
Change type	The type of change is indicated as follows:
	➤ Added. New functionality was added.
	➤ Removed. Functionality was removed.
	➤ <b>Updated.</b> 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.
	<b>Default:</b> 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 <b>Affected Part Type</b> is <b>Operation</b> and <b>Change Type</b> is <b>Updated</b> ), the change description contains details of the argument that was added or removed from the operation.

UI Elements(A-Z)	Description
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:
	➤ <b>High.</b> 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.

# 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	Changes tab > 🚉 Create Test Set
Relevant tasks	"Create regression tests" on page 160
Wizard map	The Create Change Impact Test Set wizard contains:
	Coverage Page > Test Set Page

### **Chapter 10 •** Change Analysis

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

UI Elements (A-Z)	Description
	<b>Select Columns.</b> Enables you to select the columns to display in the Changes grid.
5	Refresh. Reloads the list of changes.
Select All	Selects all records in the Changes list.
<changes grid=""></changes>	A list of all the changes for the selected component. To set the columns to display, use the Select Columns button.
	<b>Default:</b> Subject, Affected Part Type, Affected Part, Change Type, Risk, Change Status. Creation Date, Assigned To, Release, and Cycle.
Scope	The scope of the test set:
	➤ All tests. Create a test set using all tests linked to the component.
	> Tests affected by the following changes. Create a test set only using the tests that were affected by the selected change(s). Select the changes you want to analyze. Make a multiple selection using the Ctrl key.
Coverage	The components to cover with the test set. Either of the following options:
	<pre>&gt; <component_name> only &gt; <component_name> and all Dependent Components</component_name></component_name></pre>



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:
	Create Change Impact Test Set Wizard > Test Set Page

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

UI Elements (A-Z)	Description
₹ Select	<b>Select</b> . Opens the Test Plan tree to enable you to select tests to include in the test set.
×	<b>Delete.</b> Removes the selected test from the Tests grid.
1	Browse. Opens the Select Test Set folder dialog box.
1	<b>Tip:</b> In the Select Test Set folder dialog box, click the <b>New Folder</b> button □ to create a new folder.
<b>⇔</b>	<b>Add to Coverage,</b> Adds the selected test to the test grid in the left pane.
	<b>Note:</b> Only visible from the Test Plan pane.
50	<b>Refresh.</b> Reloads the list of tests in the Test Plan tree.
3	<b>Note:</b> Only visible from the Test Plan pane.
7	<b>Set Filter/Sort.</b> Enables you to set a filter for displaying the tests by a criteria.
	<b>Note:</b> Only visible from the Test Plan pane.
<tests grid=""></tests>	A list of all the tests in the test plan. To add tests, click the <b>Select</b> button and drag in tests from the Test Plan tree.

### **Chapter 10 •** Change Analysis

UI Elements (A-Z)	Description
Folder	The folder name for the test set. Use the <b>Browse</b> button to locate a folder.
Name	A name for the test plan.
Test Plan list	An expandable list of the tests available for the component.  Note: Only visible when you click the Select button.

# 11

# Reports and Graphs

This chapter includes:

### Concepts

➤ Service Test Management Reports and Graphs Overview on page 171

#### **Tasks**

- ➤ How to Create and Generate Reports on page 181
- ➤ How to Generate Graphs on page 183

# **Concepts**



# Service Test Management Reports and Graphs Overview

Quality Center'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 *Quality Center User Guide*.

This section describes the Excel reports that you can generate through Quality Center, 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.

#### **Chapter 11 • Reports and Graphs**

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 172
- ➤ "Application Component Query Builder Entities" on page 174
- ➤ "Application Component Graphs" on page 175
- ➤ "Application Changes Graphs" on page 178

### **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.

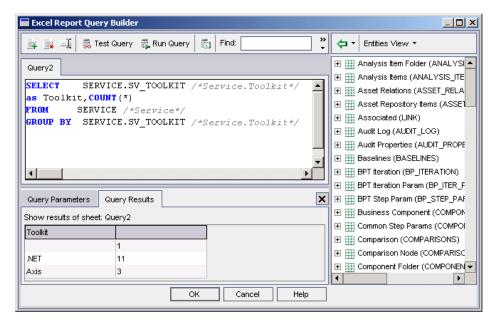
Report Name	Description
Web Service Components by Toolkit	Exports the distribution of toolkits which are used by the Web services.
Web Service Components by Type	Exports the distribution of Web service import types: File, URL, UDDI, or Systinet.

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 181.

### **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 183.

### **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**

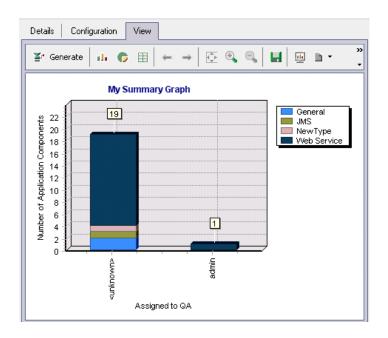
In Application Component Summary graphs, you can configure the **x-axis** to display one the following fields: Assigned to Dev, Assigned to QA, Authentication default credentials, Created by, Development status, Has changed, Import type, JMS provider, JMS transport, JNDI initial context factory, Keep 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.

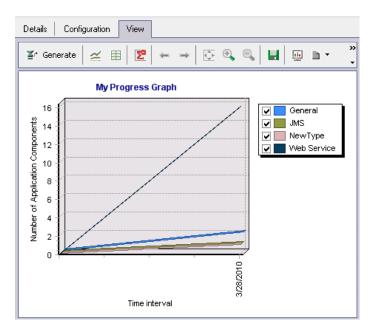
You can configure the **x-axis** to display a time interval: Last 7 days, Last 5 weeks, Last 12 months, or All days.

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.



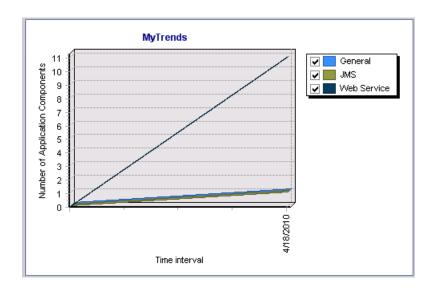
### **Trend Graphs**

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**

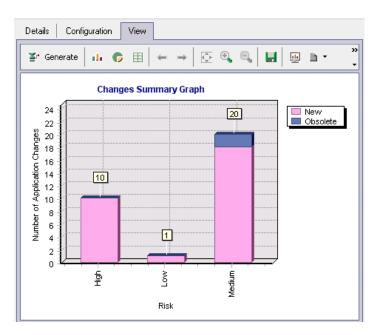
In Application Changes Summary graphs, you can configure the **x-axis** to display one the following fields: Affected part type, Assigned to, Change status, Change type, Created by, Cycle, Release, Risk, and QC Project.

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.

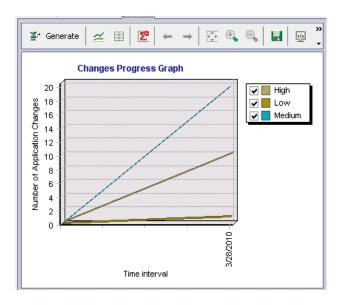
You can configure the **x-axis** to display a time interval: Last 7 days, Last 5 weeks, Last 12 months, or All days.

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.

#### **Chapter 11 • Reports and Graphs**

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**

## **P** How to Create and Generate Reports

This task describes how to create reports using Quality Center's built-in reports for application components.

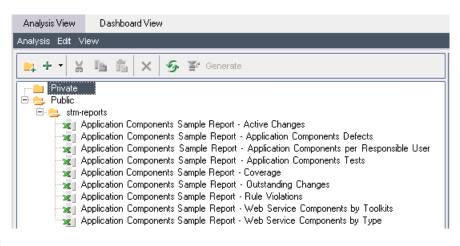
This task includes the following steps:

- ➤ "Generate a pre-defined report" on page 181
- ➤ "Create a custom report based on a pre-defined report" on page 181
- ➤ "Create new Excel reports" on page 182

#### **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.



**≚**\* <u>G</u>enerate

**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**. Quality Center 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 174.
- **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 Quality Center User's Guide*.

## How to Generate Graphs

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

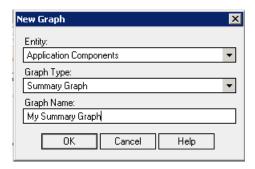
This task includes the following steps:

- ➤ "Open the New Graph dialog box" on page 183
- ➤ "Configure the graph" on page 184
- ➤ "Select different projects optional" on page 184
- ➤ "View the Graph" on page 184

#### **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).

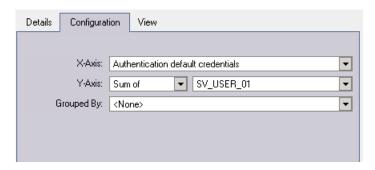


- **3** Select an Entity **Application Components** or **Application Changes**.
- 4 Select a graph type: Summary Graph, Progress Graph, or Trend Graph.

**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 175 or "Application Changes Graphs" on page 178.

## **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 Quality Center User's Guide.

# **12**

## **Custom Script Generators**

This chapter includes:

#### **Concepts**

➤ Creating Custom Script Generator Overview on page 185

#### **Tasks**

➤ Implement and Register a Custom Generator on page 186

#### Reference

- ➤ Script Generator API Reference on page 189
- ➤ Test Generation Flow on page 192

## **Concepts**



## 🖧 Creating Custom Script Generator Overview

The purpose of the custom script generator is to allow programmers to extend the capabilities of the Service Test Management Requirement and Test Generation Wizard.

Use this feature in conjunction with the custom type components, to make them more readily accessible. For details, see "Component Types Tab" on page 147.

#### **Chapter 12 • Custom Script Generators**

This API enables you to develop an external script generator through COM or .NET which can be used by Service Test Management during test generation. After you implement the custom generator, Service Test Management adds it to the list of script generators in the Requirement and Test Generation wizard.

When you run the wizard, Service Test Management invokes the API of the custom generator to fill the tests with code, according to the desired logic.

The API includes initialization of the custom generator for the specified aspects and the generation of scripts for tests associated with the component.

Tests created by the custom generator use the custom type name. The required fields are populated by default values and by user provided values. The wizard shows the required fields and their values in the first step.

The Service Test Management installation includes a sample script generator. To install this script generator, run the ScriptGeneratorExampleSetup.msi file in the in the <Service Test Management installation>/ScriptGeneratorExample folder on the server machine. This creates a new script generator titled Custom Manual Tester. This folder also contains project and source files of a .NET sample.

You can implement the script generator as a COM or .NET object. For both COM and .NET implementations, use the **IScriptGenerationTaskInfo** interface to capture generation task data when reporting generation events.

## **Tasks**



This task describes how to implement and register a custom generator. You can implement the script generator as a COM or .NET objects.

**Tip:** For a use-case scenario related to this task, see "Test Generation Flow" on page 192.

This task includes the following steps:

- ➤ "Prerequisite" on page 187
- ➤ "Capture the task data optional" on page 187
- ➤ "Implement a COM object optional" on page 187
- ➤ "Implement a .NET object optional" on page 188
- ➤ "Add the Registry key" on page 188

### **Prerequisite**

Make sure to have the Service Test Management client on the development machine, so that you may add a reference to **SOAClient** or **StmApi**. You can install the client either by accessing Service Test Management through your browser or by installing the client side installation of Quality Center and Service Test Management. For details, see the Installation guide.

## Capture the task data - optional

To capture generation task data when reporting generation events, use the **IScriptGenerationTaskInfo** interface for both COM and .NET implementations.

## Implement a COM object - optional

Link the project to the **SOAClient.dll** located in the **C:\Program Files\Common Files\Mercury Interactive\Quality Center** folder. Implement the following interfaces:

- ➤ **IScriptGenerator**. Implements the script generator mechanism.
- ➤ IScriptGeneratorCallback. Implements COM events.

#### Implement a .NET object - optional

Link the project to the **StmAPI.dll** located in the **C:\Program Files\Common Files\Mercury Interactive\Quality Center\client** folder. Implement the **INetScriptGenerator** interface.

Declare the following as COM source interfaces:

- ➤ IScriptGeneratorCallback
- ➤ The .NET object

**Note:** You must register the custom generator as a COM object, even if was implemented as a .NET object.

## **Add the Registry key**

In order for Service Test Management to include a new script generator type in the Generation wizard, you need to add it to the registry:

Add a registry key with the name of the generator, under HKEY\_LOCAL\_MACHINE\SOFTWARE\Mercury Interactive\Service Test Management\ScriptGenerators\<your\_generator\_name>.

Set the key with the following values:

Key name	Description
COM ID	A string value, set to the COM ProgID, used to create the COM object. This is the COM object that implements the above interfaces and logic.
Test Type	A string value, set to the Quality Center test type that the custom generator supports. For example, MANUAL or DB-TEST.
Name	A string value, indicating the name to display in the list of test generators in the first step of the Generation Wizard.

Key name	Description
Description	A string value, describing the custom generator. This is currently not used, but reserved for future implementations.
Supported Version	A dword value, specifying the version of Service Test Management supported by the custom generator. Set this value to 10.50.

## Reference

## **Script Generator API Reference**

The following section provides the API of the **StmAPI.dll**.

## **IScriptGenerator Interface**

The following is the object definition:

```
object,
uuid(B60C546F-EE91-48a2-A352-CFC36E613CB7),
dual,
nonextensible,
helpstring("IScriptGenerator Interface"),
pointer_default(unique)
```

This section provides the interface:

```
interface IScriptGenerator : IDispatch{
     [helpstring("Init the Script generator")]
     HRESULT Init();
   [helpstring("Set generation parameters")]
     HRESULT SetGenerationData([in] IList *SelectedAspects, [in]
SAFEARRAY(VARIANT) AdditionalData);
   [helpstring("Report how many tests are required for a given component, and a given
aspect, according to already set aspects and additional data")]
     HRESULT EvaluateTestCount([in] IService *Service, [in] IServiceAspect *Aspect,
[out] int *TestsCount);
   [helpstring("Start generation for a given service")]
     HRESULT StartGenerationForService([in] IService *Service, [in] IList *Tests);
   [helpstring("Stop all current generation")]
    HRESULT StopGeneration();
   [helpstring("General purpose error reporting")]
     HRESULT GetLastError([out] BSTR *Error);
};
```

## IScriptGenerationTaskInfo Interface

The following is the object definition:

```
object,
uuid(73C9EFD1-3B4C-4fbb-AF29-2AB59E6ABACB),
dual,
nonextensible,
helpstring("IScriptGenerationTaskInfo Interface"),
pointer_default(unique)
```

This section provides the interface:

```
interface IScriptGenerationTaskInfo : IDispatch{

[propget, helpstring("Task status message")]
    HRESULT StatusSucceeded([out] VARIANT_BOOL *Success);
[propget, helpstring("Task error message")]
    HRESULT ErrorMessage([out] BSTR *Error);
[propget, helpstring("The id of the generation aspect this task is handling")]
    HRESULT AspectID([out] int *AspectID);
[propget, helpstring("The id of the generated test this task is handling")]
    HRESULT TestID([out] int *TestID);
[propget, helpstring("The name of the generation aspect this task is handling")]
    HRESULT AspectName([out] BSTR *AspectName);
};
```

## IScriptGeneratorCallback Interface

The following is the object definition:

```
object,
uuid(BEEE5AA8-002E-4221-B99B-6E592F23E17E),
dual,
nonextensible,
helpstring("IScriptGeneratorCallback Interface"),
pointer_default(unique)
```

This section provides the interface:

```
interface IScriptGeneratorCallback : IDispatch{
    [helpstring("Generation task started")]
    HRESULT TaskStarted();
    [helpstring("Generation task ended")]
    HRESULT TaskEnded([in] IScriptGenerationTaskInfo *TaskInfo);
};
```

**Note:** The **StopGeneration** and **GetLastError** methods are currently not in use.

## Test Generation Flow

The following section describes the flow of actions Service Test Management performs while creating tests through a custom script generator:

- **1** Creates the custom generator through its COM registration (even for .NET objects).
- **2** Registers the generation events. If the custom generator is a COM object, it registers them through Advise. For a .NET object, it registers the .NET events directly.
- **3** Calls **Init** to run all initialization code before the actual generation process.
- **4** Calls **SetGenerationData** with the selected aspects as the first argument and null as the second argument. The **AdditionalData** argument is not used, and reserved for the future. The custom generator uses this call to store the selected aspects and prepare for generating scripts for these aspects.
- **5** Starts the generation process in a new thread, to keep the user interface active.
- **6** Loops through the selected component or component group. For each component, Service Test Management:
  - **a** Loops through the selected aspects and calls **EvaluateTestCount** for the component and each of its aspects. The custom generator implements this method to return the number of tests to generate for that component and aspect. This is usually one test, but it can also be a multiple number of tests.
  - **b** Creates the tests and initializes their fields' values. It posts the tests so that the custom generator receives actual, not virtual, tests.
  - **c** Calls **StartGenerationForService** for the current component and its tests for each of the aspects.
  - **d** The custom generator creates the test scripts using the OTA API or any other third party API. As these are actual tests, the custom generator can read and write to their fields, upload attachments, and so forth.

- e During script generation, the custom generator should invoke the TaskStarted event before generating each test and the TaskEnded event after completing the test generation. The status of the script generation for that test should be reported in TaskEnded TaskInfo argument (implementing the IScriptGenerationTaskInfo interface), indicating the following values:
  - ➤ Status of the task
  - ➤ Error message (if applicable)
  - ➤ Aspect ID
  - ➤ Script Name
  - ➤ Test ID
- **f** Service Test Management displays the data and stores the information to make it available for other uses, such as linking the test with requirements.
- **g** Service Test Management posts the tests again, committing any changes to their fields' values.
- h If requirements were also generated through the wizard, Service Test Management links the tests to their respective requirements.

**Chapter 12 •** Custom Script Generators

## Index

A	overview 155
Add Change task 153	Change Impact test wizard 167
Add Group task 152	change risk 165
Add Service Folder task 152	change status, Changes tab 166
Add Service task 152	change types 164
Add Services from Systinet 63	changes
Add Services from UDDI 64	automatic detection 157
Adobe Reader 12	reporting manually 156
alerts	changes alert 78
disregarding 28	Changes tab 161, 162
handling 78	change risk 165
Alerts dialog box 28	change status 166
Application Components module 48	change type 164
Application Components tab	creation date 166
Requirements Module 127	getting started 49
Test Plan Module 130	owner 163
aspects	part name 165
checking coverage 110	part type 165
coverage 115	Component name, Details tab 30
testing 91	components
Aspects tab	alerts 78
customization 135	associating to group 84
Testing Status 115	copying 79, 81
associate components 84	dependencies 23
Attachments tab 49	detail management 78
Axis/Java based Web services 91	find 78
	linking to requirements, tests 123
D	linking, general 125
В	mailing 82
Boundary testing	managing 77
extreme values 92	requirements and tests 89
null values 92	searching for 78
	types 147
C	Connection Settings dialog box 66
-	contract
change analysis 158	changes 155
contract changes 155	download 52

## Index

file location 33	Details tab 29
copying components 81	direct links in Quality Center 26
coverage 103	Documentation Library 11
aspects 115	download contract 52
dashboard 113	
defects 120	-
defects, overview 109	E
group 85	empty test 96
operations 107, 118	Emulate Component task 153
overview 104	emulation of Web services 61
requirements 106, 116	entities
rules 104, 114	project 148
tests 119	report Query builder 174
tests, overview 108	1 ,
Create Change Impact Test wizard 167	F
creation date, Changes tab 164	•
cross-site scripting (XSS) 91	Find dialog box 35
cross-site scripting, testing 91	finding components 78
custom script generator 185	
customization	G
General tab 141	_
Groups pane 151	General tab, customization 141
Rules tab 143, 145	generating
Customize Component task 153	graphs 183
customizing 133	reports 181
component types 147	tests and requirements 94
module access 154	generic toolkit (deprecated) 91
overview 133	Getting Started 12
	graphs 171
project entities 148	graphs, generating 183
projects 138	group coverage 85
	groups
D	associating components 84
dashboard, coverage 113	component 83
defects	manage 85
coverage 109, 120	overview 83
coverage, viewing 112	user interface 87
Defects tab 120	Groups pane, customization 151
Delete Change task 153	
Delete Component Folder task 152	н
Delete Component task 152	
Delete Group task 153	Help on this page command 11
	Host Selection dialog box 75
dependencies between components 23	
setting details 42	1
	-
Dependency Details dialog box 42	Import Service dialog box 68

importing services 68 importing, Web Services 57 Interaction tab 32 interoperability testing .NET Framework 91 Axis/Java based Web services 91	coverage overview 107 coverage, viewing 110 Operations tab 118 Overload testing 92 Owner, Changes tab 165
generic toolkit (deprecated) 91	P
IScriptGenerationTaskInfo 190	
IScriptGenerator 189	Part Name, Change tab 163 Part Type, Change tab 163
IScriptGeneratorCallback 191	Performance testing
	longevity 92
J	overload 92
JMS test settings 45	scalability 92
JNDI 34	stress 92
JNDIU definitions 45	volume 92
	pivot table 174
L	Point to point 34
Limb Coming to Demain and took 150	Positive testing 91
Link Service to Requirements task 152 Link Test to Requirement 126	Publish and subscribe 34
linking to requirement, tests 123	
linking, components to tests 124	Q
longevity testing 92	Quality Center
0 , 0	direct links 26
M	Documentation Library 11
	entities 19
mailing services 82	integration 16
Manual Runner, testing services 71	Service Test Management extension 9
Modeling tab 36 Modify Change task 153	Query builder, for Excel report 174
Modify Component Folder task 152	
Modify Component task 152	R
Modify Group task 153	rage 105
module access 154	Readme 12
	registry, Systinet 16
N	reports, generate and create 181
	required fields 148
New Change dialog box 163	requirements
New Component dialog box 43 New Folder dialog box 43	coverage 116
New Group dialog box 87	coverage observing 110
Then Group dialog box of	coverage, checking 110 Requirements tab 116
	rules
0	coverage 114
operations	violations 104
coverage 118	Rules tab 114, 143, 145

Run operation dialog box 69 toolkit support 57 S U Scalability testing 92 UDDI, adding from 64 script generator, custom 185 Update Component dialog box 159 flow 192 Update Component/Group task 152 implement 186 overview 185 V search, for components 78 Volume testing 92 Security testing cross-site scripting 91 SQL injection vulnerability 91 W service emulation 61 Web Service call 58 Service Test Management, overview 15 Web services 55 Service Test, creating scripts 59 address 33 spot testing 58 defining 60 SQL injection vulnerability 91 emulation 61 Standard Compliance testing 91 importing 57, 68 Stress testing 92 What's New command 12 Systinet registry 16 WSDL. See contract Systinet, adding from 63 X т XSS scripting 91 test generation flow 192 Test Plan module, components 130 Test Set wizard 167 select test set 169 testing aspects 91 Testing Status tab 112 Application Components module 49 Aspects tab 115 Dashboard tab 113 Defects tab 120 Operations tab 118 Requirements tab 116 Rules tab 114 Tests tab 119 tests coverage 119 coverage, viewing 111 creating empty 96 generating 94 Test Plan module 130 Tests tab 119