

# HP Service Test

for the Windows operating system

Software Version: 11.00

---

## User Guide

Document Release Date: October 2010

Software Release Date: October 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.

### Copyright Notices

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

### Trademark Notices

Java™ 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](http://h20230.www2.hp.com/new_access_levels.jsp)**

---

# Table of Contents

<b>Welcome to This Guide</b> .....	<b>9</b>
How This Guide Is Organized .....	9
Who Should Read This Guide .....	10
Service Test Updates .....	10
Service Test Documentation.....	11
<b>Chapter 1: Creating and Saving Tests</b> .....	<b>13</b>
<b>Concepts</b> .....	<b>14</b>
Test Overview .....	14
<b>Tasks</b> .....	<b>15</b>
How to Create a New Test .....	15
How to Define Test Properties or User/System Variables.....	16
<b>Reference</b> .....	<b>19</b>
Test User Interface .....	19
<b>Chapter 2: Working with Service Test</b> .....	<b>21</b>
<b>Concepts</b> .....	<b>22</b>
Service Test Overview .....	22
UFT (Unified Functional Testing) .....	36
JMS Transport Overview.....	36
<b>Tasks</b> .....	<b>41</b>
How to Populate a Test.....	41
How to Implement UFT testing .....	47
How to Create a Call to a Java Class .....	49
<b>References</b> .....	<b>53</b>
The Service Test User Interface.....	53
Security Scenarios User Interface.....	84

<b>Chapter 3: Defining Services</b> .....	<b>97</b>
<b>Concepts</b> .....	<b>98</b>
Preparing Service References .....	98
Setting Security Overview .....	98
Security Scenarios Overview.....	99
WCF Scenario Settings .....	103
Advanced Security Settings .....	109
Negative Testing.....	114
<b>Tasks</b> .....	<b>116</b>
How to Import a WSDL-Based Web Service.....	116
How to Set Security for a Web Service .....	118
How to Create a Simple Test for a REST Service .....	122
How to Create a Template-Based REST Service .....	125
How to Set up Common Web Services Security Scenarios .....	129
<b>Reference</b> .....	<b>133</b>
Import Service User Interface.....	133
Security User Interface.....	136
<b>Chapter 4: Data Handling</b> .....	<b>147</b>
<b>Concepts</b> .....	<b>148</b>
Data Handling Overview.....	148
Defining Data .....	148
Data Handling .....	153
<b>Tasks</b> .....	<b>158</b>
How to Create Data Queries.....	158
How to Assign Data to Test Steps.....	161
How to Set the Navigation Properties .....	164
How to Data Drive a Test Step .....	165
<b>Reference</b> .....	<b>167</b>
Data Handling User Interface.....	167
<b>Chapter 5: Running Tests</b> .....	<b>183</b>
<b>Concepts</b> .....	<b>184</b>
Running Tests Overview .....	184

<b>Tasks</b> .....	<b>187</b>
How to Run a Test .....	187
How to Prepare and Run a Load Test .....	190
<b>Reference</b> .....	<b>194</b>
Command Line Syntax .....	194
Run Test User Interface .....	195
Debug Tabs User Interface.....	197
<b>Chapter 6: ALM/QC Integration</b> .....	<b>201</b>
<b>Concepts</b> .....	<b>202</b>
Managing Tests Through HP ALM/QC Overview .....	202
ALM/QC Tests and Version Control .....	203
<b>Tasks</b> .....	<b>204</b>
How to Work with Tests in ALM/QC.....	204
How to Save Tests to ALM/QC Projects .....	206
How to View a List of Previous Test Versions .....	207
<b>Reference</b> .....	<b>208</b>
ALM/QC User Interface .....	208
<b>Chapter 7: Run Results Viewer</b> .....	<b>211</b>
<b>Concepts</b> .....	<b>212</b>
Run Results Viewer Overview.....	212
Run Results File Location .....	214
Captured Data .....	215
Custom Fields .....	218
<b>Tasks</b> .....	<b>220</b>
How to Open Run Results .....	220
How to Navigate the Run Results Tree.....	221
How to Manually Submit Defects to Quality Center.....	222
How to Export Run Results .....	224
<b>Reference</b> .....	<b>226</b>
Run Results Viewer User Interface.....	226
Run Results Viewer Commands .....	229
Run Results Viewer Panes.....	232
Run Results Viewer Dialog Boxes .....	245

<b>Appendix A: Coding Service Test Events .....</b>	<b>261</b>
<b>Concepts .....</b>	<b>262</b>
Writing Code for Events Overview .....	262
<b>Tasks .....</b>	<b>264</b>
How to Open a Window for Writing Custom Code .....	264
How to Access Input and Output Properties.....	265
How to Use the Logging Function .....	266
How to Use the Report Function .....	267
How to Write Checkpoint Events .....	268
How to retrieve and set test variables .....	270
How to encrypt and decrypt passwords.....	270
How to use the OnReceiveResponse Web Service Event .....	272
<b>Reference .....</b>	<b>274</b>
Context methods.....	274
Logging Options.....	275
Assert Options .....	276
<b>Index .....</b>	<b>277</b>



---

# Welcome to This Guide

Welcome to HP Service Test, HP's tool for creating tests, with a focus on testing SOA and headless technologies.

**This chapter includes:**

- ▶ [How This Guide Is Organized](#) on page 9
- ▶ [Who Should Read This Guide](#) on page 10
- ▶ [Service Test Updates](#) on page 10
- ▶ [Service Test Documentation](#) on page 11

## How This Guide Is Organized

This guide contains the following chapters:

**Chapter 1**    **Working with Service Test**

This chapter describes the Service Test interface, its layouts, and dockable window panes.

**Chapter 2**    **Creating and Saving Tests**

This chapter describes how to create a new solution and project.

**Chapter 3**    **Defining Services**

This chapter describes how to import services and maintain them.

## Chapter 4 Data Handling

This chapter describes how to add data to your test and how to assign it to your test's parameters.

## Chapter 5 Running Tests

This chapter describes how to run tests from the user interface or command line.

## Chapter 6 Run Results Viewer

This chapter describes how to view the results in the Run Results Viewer and filter or export the information according to your needs.

## Chapter 7 ALM/QC Integration

This chapter describes how to integrate with HP ALM/QC (Application Lifecycle Management/Quality Center) to manage your tests and implement version control.

## Who Should Read This Guide

This guide is for the following users of Service Test:

- ▶ Application developers or configurators
- ▶ System or instance administrators
- ▶ Functional and load testers

Readers of this guide should be moderately knowledgeable about SOA systems and Web services.

## Service Test Updates

You can check online for updates to all HP products installed on your computer at any time by choosing **Start > Programs > HP Service Test 11.00 > HP Update**. You can then choose which updates to download and (optionally) install.

## Service Test Documentation

Service Test includes the following online documentation:

**Readme.** Choose **HP Service Test > Readme** from the Start menu.

**HP Service Test Online Help.** Available from the Service Test user interface by clicking in the window and pressing F1 or clicking the **Help** button.

**User Friendly Documentation.** Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site ([www.adobe.com](http://www.adobe.com)).

- ▶ **HP Service Test User Guide.** Choose **HP Service Test 11.00 > Documentation > User Guide** from the Start menu.
- ▶ **HP Service Test Installation Guide.** Choose **HP Service Test 11.00 > Documentation > Installation Guide** from the Start menu.
- ▶ **HP Service Test Tutorial.** Choose **HP Service Test 11.00 > Documentation > Service Test Tutorial** from the Start menu.

Welcome to This Guide

# 1

---

## Creating and Saving Tests

This chapter includes:

### Concepts

- ▶ [Test Overview](#) on page 14

### Tasks

- ▶ [How to Create a New Test](#) on page 15
- ▶ [How to Define Test Properties or User/System Variables](#) on page 16

### Reference

- ▶ [Test User Interface](#) on page 19

---

---

## Concepts

---

---

### Test Overview

To test your applications, you use Service Test to create tests.

You can base the new test on the standard Service Test template, or, if you have LoadRunner installed, on a Load Testing template.

You can create tests on either the file system, or within HP ALM/QC (Application Lifecycle Management/Quality Center).

---

---

## Tasks

---

---

### How to Create a New Test

The following steps describe how to create a new test. Each test provides an empty canvas to which you can add activities.

- "Set the path type - optional" on page 15
- "Open the create a new test dialog box" on page 15
- "Define test properties" on page 16
- "Define user variables" on page 17

#### 1 Set the path type - optional

By default the path used for the test is relative. If you plan to transfer the test to other machines, you can use an absolute path. Select **Edit > Settings** and clear the **Use a Relative Path** option.

#### 2 Connect to ALM/QC- optional

To create tests in an ALM/QC repository, first connect to the ALM/QC server. For details, see "How to Connect to ALM/QC" on page 206.

#### 3 Open the create a new test dialog box

Click or **File > New > Test**. For user interface details, see the "Create New Test Dialog Box" on page 19.

#### 4 Select a template

In the Create a New Test dialog box, select the **Service Test** template. For load testing, select the **Service Test Enabled for Load Testing** template. The latter template is only available when LoadRunner is installed on the machine.

## 5 Specify a name

Accept the default name or specify your own string in the **Name** box.

## 6 Select a location

Accept the default location, or click the **Browse** button adjacent to the **Location** box to open the Open Folder dialog box.

For a location on the file system, browse to the folder. and click **Open**.



For ALM/QC tests, click the **ALM/QC Test Plan** button in the left pane, and navigate to the folder. Use the New Folder .button to create a new branch. Click **Open**.

## 7 Generate the test

In the Create a New Test dialog box, click **Create**.

# How to Define Test Properties or User/System Variables

The following steps describe how to define test properties, user variables, and operating system variables. These settings are optional.

- "Define test properties" on page 16
- "Define user variables" on page 17
- "Define user variables profiles" on page 18
- "Set user variable values" on page 17
- "Set operating system variables" on page 18

## Define test properties

This step applies if you want to define custom properties that can be used by all steps in the test, with the ability to assign data from a query.



- 1 Click in a blank area of the canvas. In the Property Sheet, open the **Test Input Parameters** view.
- 2 Click the **Add Property** button to define a new input or output property.



- 3 In the **Add Input/Output Property** dialog box, specify a property name and type.
- 4 Click in the **Default Value** column. Specify a value or click the **Link to Source** button to specify a data source. For details, see "Select Link Source Dialog Box" on page 174.

### Define user variables

Service Test lets you create user variables and set their values. You can define multiple profiles for the variable values. You select a profile to be active before the test run. For details, see the next step.

- 1 Click in a blank area of the canvas. In the Property Sheet, open the **Test Variables** view.
- 2 Click the **Add User Variable** button to define a new user variable.

### Set user variable values

Set values for the variables in one of the following ways:

- ▶ In the Property Sheet's **Test Variables** view, click in the **Profile** column and manually enter values.
- ▶ Open the Toolbox (left pane) and drag the **Set Test Variable** activity from the **Miscellaneous** category into a loop, such as Test Flow. In the Property Sheet, define the variable key and value. To obtain values, click the **Link to a Data Source** button in the row. In the Select Link Source dialog box, select the Test Variables option. Select a name and value for the **Variable key** and **Variable value**.
- ▶ Open the Property Sheet's **Events** view for the step or define a Custom Code activity. Edit the Event Handler code and assign a value using the **TestProfile** object. The following example sets the value of *Region* user variable to NE.

```
activity.Context.TestProfile.SetVariableValue("Region", "NE");
```

For details, see Appendix A, "Coding Service Test Events."

Repeat this step for each variable for which you want to set values.

## Define user variables profiles

This step only applies if you created user variables as described in the previous step.

**1** Define one or more user variables as described above.



**2** Click the **Add New Profile** button.

**3** In the New Test Profile dialog box, specify a name and indicate the profile (if any) from which to copy the properties. If you do not copy from an existing profile, Service Test copies the user variables from the active profile without its values.



**4** To rename or delete a profile, click the **Manage Profiles** button. Click **Remove** or **Rename**.



**5** Click the **Compare** button to display the profiles side-by-side.

**6** Only the active profile is accessed during the test run. To make a profile active, select it from the **Active Profile** dropdown list.

For user interface information, see the "Test Variables View" on page 77.

## Set operating system variables

**1** Open the Toolbox (left pane) and drag the **Set OS Environment Variable** activity from the **System** category into the loop, such as Test Flow.



**2** In the Property Sheet's **Input/Checkpoints** view, define the variable key and value or click the **Link to Source** button to specify a data source.



---

**Tip:** To view a list of the test variables, click in a blank area of the canvas. In the Property Sheet, open the **Test Variables** view. The System variables are listed in the lower pane.

---

---



---

## Reference

---



---

### Test User Interface

This section includes:



- ▶ [Create New Test Dialog Box](#) on page 19

### Create New Test Dialog Box

This dialog box enables you to specify the location and name of the new test.

<b>To access</b>	<b>File &gt; New &gt; Test</b> (Ctrl + Shift +N)
<b>Relevant tasks</b>	"Open the create a new test dialog box" on page 15

User interface elements are described below:

UI Elements (A-Z)	Description
	<b>Browse.</b> Selects a location for the new test. To select a location on an ALM/QC server, first connect to the server. For details, see "How to Connect to ALM/QC" on page 206.
	Creates a new test in the selected location.
<b>Categories pane</b>	A list of the test categories.
<b>Location</b>	The complete path of the test. Use the <b>Browse</b> button to choose a location.

UI Elements (A-Z)	Description
<b>Name</b>	A name for the test. May contain only letters, numbers, underscores, spaces, and dots (".").
<b>Templates pane</b>	A collection of templates. <b>Service Test</b> and <b>Service Test Enabled for Load Testing</b> . <b>Default value:</b> Service Test <b>Note:</b> The <b>Service Test Enabled for Load Testing</b> template is only visible when HP LoadRunner is installed.

# 2

---

## Working with Service Test

This chapter includes:

### Concepts

- ▶ [Service Test Overview](#) on page 22
- ▶ [UFT \(Unified Functional Testing\)](#) on page 36
- ▶ [JMS Transport Overview](#) on page 36

### Tasks

- ▶ [How to Populate a Test](#) on page 41
- ▶ [How to Implement UFT testing](#) on page 47
- ▶ [How to Create a Call to a Java Class](#) on page 49

### References

- ▶ [The Service Test User Interface](#) on page 53
- ▶ [Security Scenarios User Interface](#) on page 84

---

---

## Concepts

---

---

### **Service Test Overview**

Welcome to HP Service Test, HP's tool for the construction and execution of functional tests for headless systems.

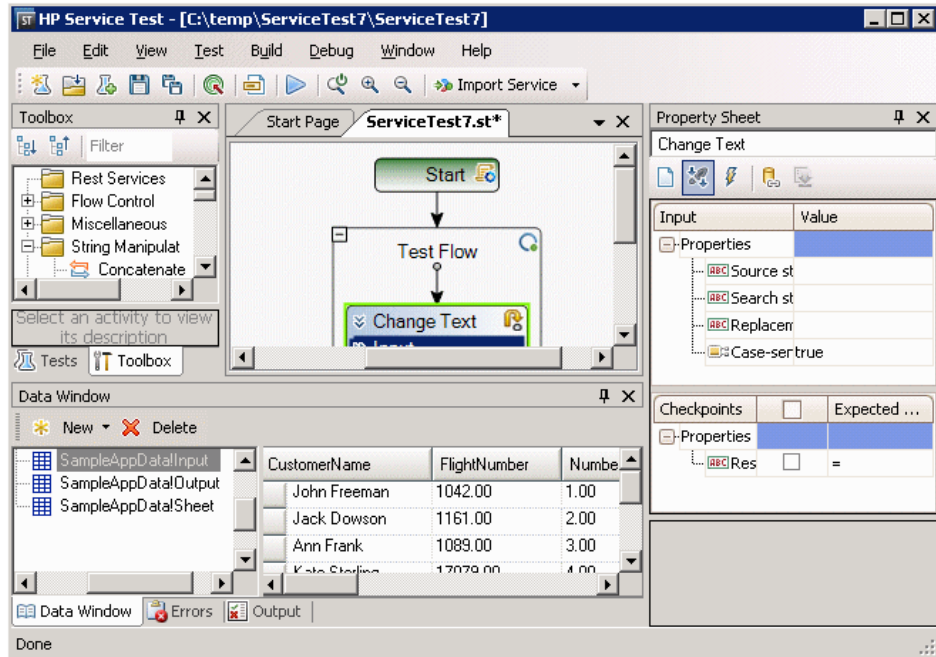
You create a test by dragging and dropping activities from the Service Test toolbox into a canvas. The toolbox provides a collection of activities for functional testing in areas such as REST, Web Services, JMS, and HTTP. You can add more activities to the toolbox by importing WSDLs or providing other contract definitions.

This section includes:

- ▶ "Introducing Service Test" on page 23
- ▶ "Performing Integrated Testing" on page 23
- ▶ "The Canvas" on page 24
- ▶ "The Toolbox" on page 29
- ▶ "Data Sources" on page 29
- ▶ "Using Service Test for Testing Web Services" on page 31
- ▶ "Using Service Test for Testing REST Services" on page 31
- ▶ "Java Testing" on page 32
- ▶ "The Property Sheet" on page 33

## Introducing Service Test

Service Test uses a visual designer as a canvas for creating steps in a test. You create tests by dragging activities from a toolbox palette into the canvas and defining their parameters.



For integrated testing, Service Test supports a variety of activities such as string manipulation, file verifications, and flow control. The toolbox also provides access to Web and REST Service operations and methods.

## Performing Integrated Testing

In testing your systems, you perform known activities and observe the response. By checking the response, you determine whether your system is performing as expected.

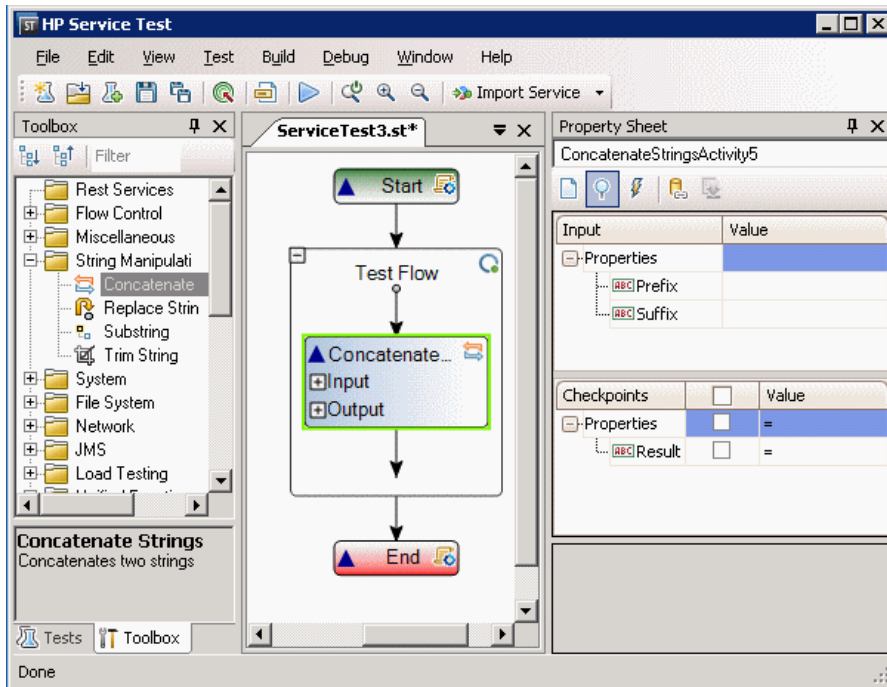
For example, in a basic **Replace String** activity, you provide the old text, new text, and an expected value. You compare the response with the expected value to confirm that your activity functioned properly.

The steps to create a basic integrated test are:

- ▶ Drag an activity into the canvas
- ▶ Assign input data
- ▶ Set checkpoints to verify the response

## The Canvas

The canvas in Service Test provides a visual representation of the test flow. You populate the canvas by dragging in activities from the Toolbox palette.



You can drag Web Services, REST services, or any Toolbox activity, into the canvas. For a service to appear in the Toolbox, you must first import it (Web services) or define it (REST services) in your project. For details, see "How to Import a WSDL-Based Web Service" on page 116.



### Tab Title

The canvas tab title is <Test name>.st. An asterisk denotes changes that were not yet saved.

### Manipulating Steps

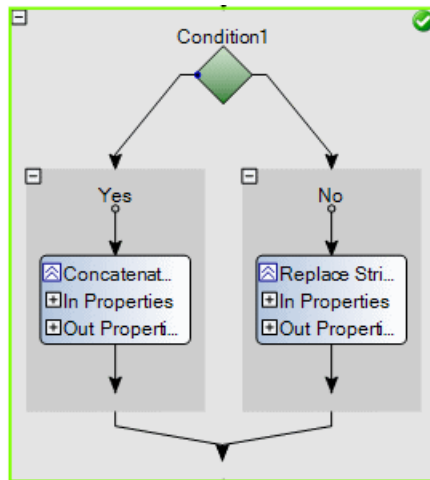
The canvas allows you to manipulate the steps in the following ways:

- **Reorder.** Drag a step from one location to another.
- **Copy and Paste.** Select a step and press **Ctrl+C** to copy it to the clipboard. Press **Ctrl+V** to paste it into another location within the Test Flow or to another loop.
- **Delete.** Select a step and press the keyboard's **Delete** button.

### Flow Control

The canvas allows you to control the flow of the steps in the following ways:

- **Loop.** The main **Test Flow** area provides a default loop for test steps. You can add steps to the Test Flow or add additional loops. Using the Properties Sheet, you can specify the type of loop, number of iterations, and conditions for the Test Flow or loop. For details, see "Input/Checkpoints View" on page 70.
- **Conditional steps.** Adds a two-branched conditional node to the test.

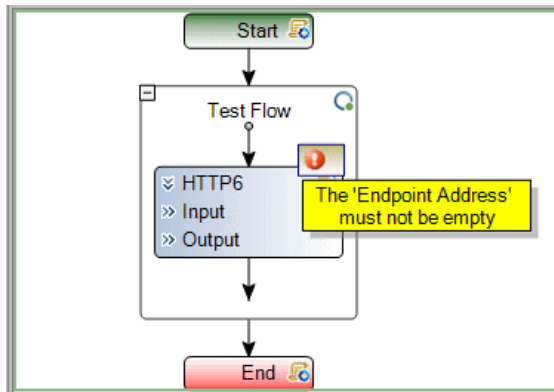


For conditional steps, you drag the **Condition** activity from the Toolbox into the canvas, and then drag an activity into each of the conditional branches.

- ▶ **Break.** Moves control to the step after the loop.
- ▶ **Continue.** Moves control back to the first step in the **Test Flow** or **Loop** and increments it to the next iteration.

### Alerts

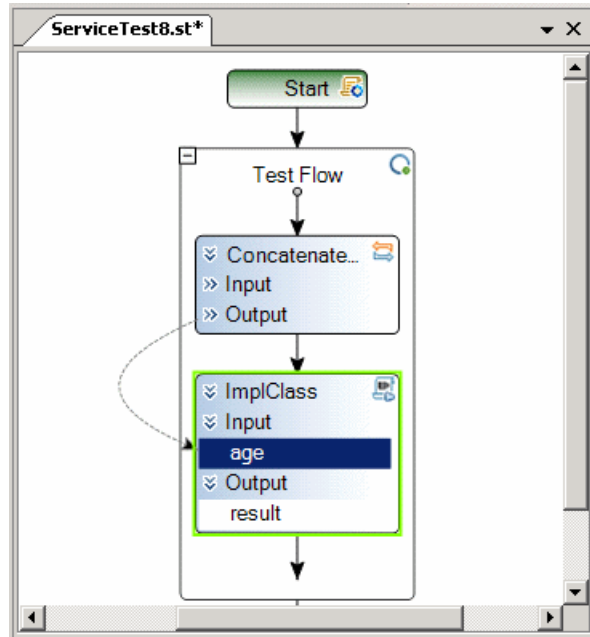
Steps that require intervention in order to run, such as **HTTP**, **Custom Code**, and **SOAP Request**, post an alert in their top right corner. Clicking an alert button displays the action that you are required to do.



You can also view this information in the Errors tab by selecting **View > Errors**.

### Connecting Data

The canvas displays the link between output parameters and input parameters from a previous step. An arrow indicates a relationship between the parameters.

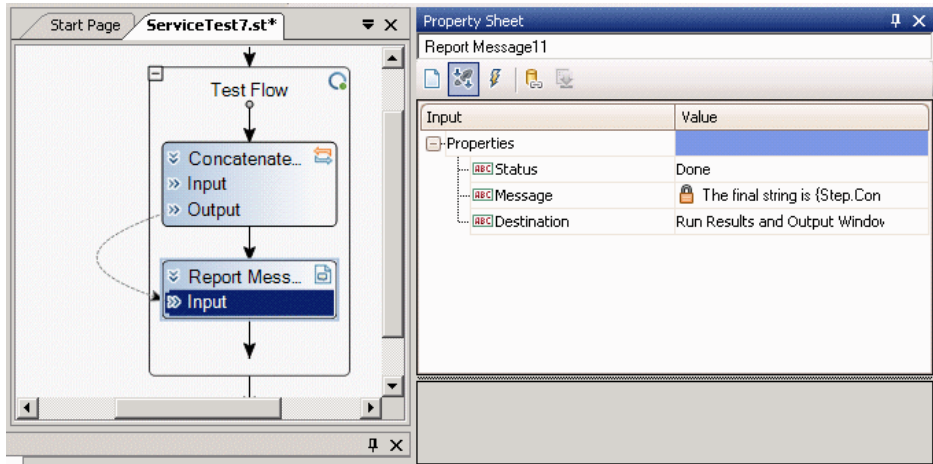


For more information, see "Data Handling User Interface" on page 167.

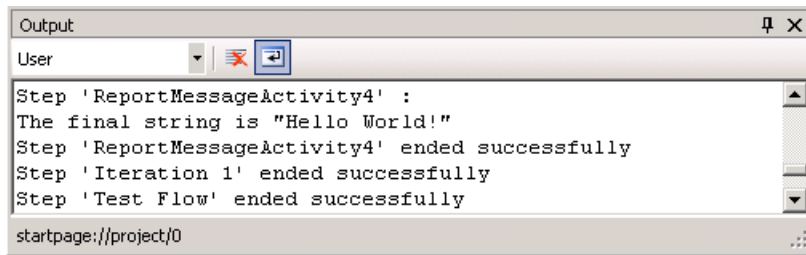
### **Sending Messages to the Output and Results Viewer**

The **Report Message** activity lets you send custom messages to the Output log and/or the Results viewer. You drag the activity to the canvas and set its properties in the Property sheet.

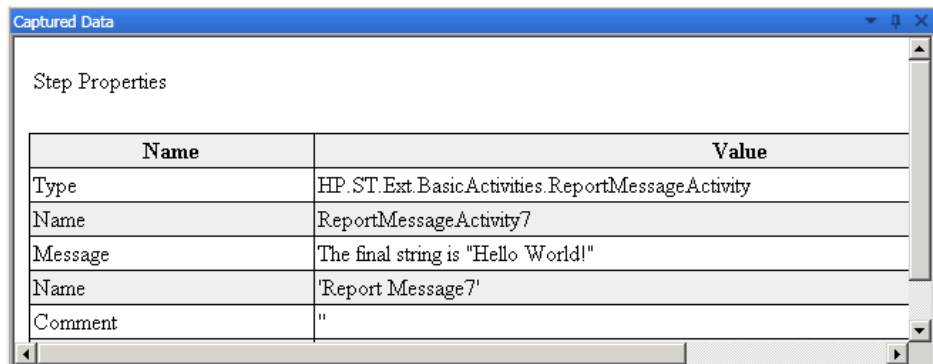
As with any step, you can link its input to the output of a prior step and show the results in the Output window.



In the following example, the Output window shows the message string.



When you use this activity, Service Test adds a **Message** row to the Results report, with the contents of the message.

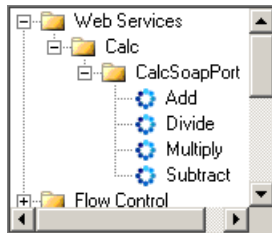


For more information, see the "Run Results Viewer" on page 211.

## **The Toolbox**

The Toolbox palette contains all of the activities and flow control objects that you need in order to create a test.

After you import a Web service, it appears in the Toolbox, under the **Web Services** node. When you expand the Web Services node, the Toolbox displays the service's name, port, and operations.



For additional details about each of the nodes, see the "Toolbox Palette" on page 59.

## **Extensibility**

For greater control over test steps, you can define custom code activities that seamlessly integrate with Service Test. You can also customize the behavior of existing activities using event handlers. For more information, see "Coding Service Test Events" on page 261.

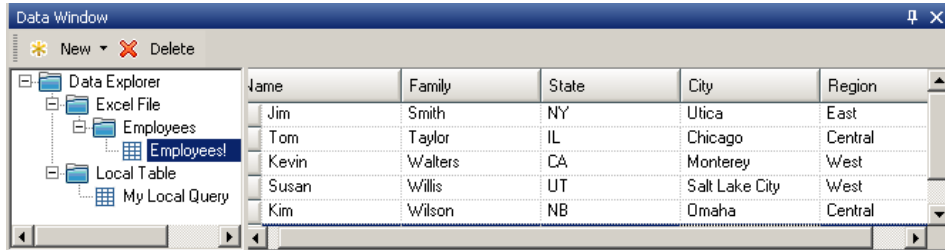
For information about adding new activities, speak with your HP representative.

## **Data Sources**

Using the Data window, you can define data for your test's properties. You can link to a variety of data sources, such as Excel or XML files. You can also define data locally for the test steps.

When you load data from Excel, you can indicate whether to make a local copy of the data, or a referenced query which refers to the data at its original source.

A data grid displays the Excel and locally defined data. If you created a referenced query, you will be unable to edit the values in the data grid.



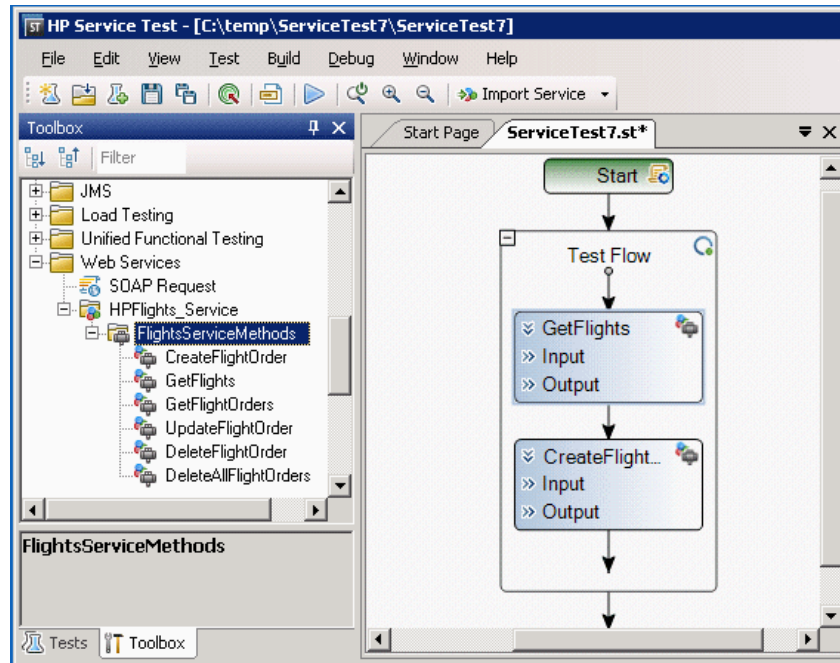
After you load data, you can assign it to properties in your test.

For information on using data in your test, see "How to Assign Data to Test Steps" on page 161.

## Using Service Test for Testing Web Services

You can use Service Test to test standard Web Services. For Web services, you import the service's WSDL file.

After importing the WSDL file, the Toolbox window lists the operations in the Toolbox. You then drag the operations directly into canvas and provide input values.



For task details, see "How to Import a WSDL-Based Web Service" on page 116.

## Using Service Test for Testing REST Services

You can use Service Test to test non-SOAP Web Services, such as REST.

For REST services, you manually define the metadata and model your activities based on the design of your service's requests.

After setting the REST resource definitions, you can create REST method calls based on the metadata. To validate the results, you set checkpoints for the REST method's response.

Service Test allows you to create templates for individual REST methods. The templates include all of the request and header information, as well as custom parameters. Using this template, you can drag the method into the canvas without having to configure its details repeatedly.

For task details, see "How to Create a Template-Based REST Service" on page 125.

## Java Testing

Service Test allows you to call Java classes with its **Call Java Class** activity.

In order to use this step, you must implement the Service Test Java interface. The aim of this interface is to define a contract or bridge between the Java artifacts owner and Service Test. The interface contains three methods:

- ▶ **getInputProperties.** Returns a mapping of the input property names and their Java class.
- ▶ **getOutputProperties.** Returns a mapping of the output property names and their Java class.
- ▶ **Execute.** A method that receives the mapping of the input properties names and their actual values i.e. their object instance. In this method, you process input properties and delegate them to your own Java artifacts. Afterwards you process the output properties and send their mappings and their actual values as the method's output.



The following table lists the relevant paths:

File	Path
<b>Interface code</b>	<ul style="list-style-type: none"> <li>▶ <b>Path:</b> %ST_DIR%\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java\ServiceTestCall.java</li> <li>▶ <b>Default installation:</b> C:\Program Files\HP\Service Test 11\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java\ServiceTestCall.java</li> </ul>
<b>Sample code</b>	<ul style="list-style-type: none"> <li>▶ <b>Path:</b> %ST_DIR%\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java\sample</li> <li>▶ <b>Default installation:</b> C:\Program Files\HP\Service Test 11\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java\sample</li> </ul>

To configure a Java step, you select a root path, a jar file containing the class, and the actual Java class. You can provide additional classpaths and jar files for the Java call. For details about selecting the Java class and packaging, see the "Call Java Class Settings Dialog Box" on page 95.

Service Test supports the following Java classes: **Byte**, **Short**, **Int**, **Long**, **Float**, **Double**, **Boolean**, **Date**, and **String**. The **Char** class is not supported.

For task details, see "How to Create a Call to a Java Class" on page 49.

## The Property Sheet

The Property Sheet provides several views that allow you to set properties for each of the test steps.

### General View



The **General** view shows the **Step ID** and **Name** properties of the test step. Several activities have additional General properties. For example, for Web Service operations, this view also shows the **Endpoint Address**, **SoapAction**, **ContentType** and **IsOneWay** setting. For details, see the "General View" on page 68.

## Input/Checkpoints View



The Input/Checkpoints view displays a grid of properties and their values. This view shows properties that apply to the selected activity or step:

Selected step	Title	Displayed in view
Activities	Input/Checkpoint	Input and checkpoint properties and their values.
Canvas, Start step, End step	Test Input Parameters	Input and Output properties for the test.
Loop (Test Flow or additional loops)	Input	The type of loop and the relevant properties such as the number of iterations. For details, see "Loop Settings" on page 72.

The display grid shows important information about each of the elements such as whether they will be encoded as base64 or whether they can be set to a NIL value. A green triangle indicates an optional element. The element is included when the toggable triangle is filled and excluded when it is cleared. For a list of the icons in the grid, see the "Input/Checkpoints View" on page 70.

You can automatically select data for the properties from data sources, system variables, and so forth. For details, see "Select Link Source Dialog Box" on page 174.


## Events View




The **Events** view lets you select or set a handler function for each of the step's events, or to open an editor to write code describing what to do when the event occurs.

For information on creating an event handler, see "Specify an event handler - optional" on page 45.


## Test Variables View

 The **Test Variables** view, is available for the **Start** and **End** steps and when standing on an empty part of the canvas. It displays the user and system-defined test variables. You can save a set of user variables in a **profile**. You can create multiple profiles and compare them. For details, see "Define user variables" on page 17.


## REST HTTP Input View

 The **REST HTTP Input** view, available when selecting a REST service method in the toolbox, stores the **Endpoint Address**, **HTTP Method**, **HTTP Version**, and the **Request Headers**. For details, see the "REST HTTP Input View" on page 78.


## HTTP View

 The **HTTP** view, available for the HTTP activity and REST service methods, displays the HTTP request, response and version. You can also set the headers and import the body of the request. For details, see "HTTP View" on page 78.


## SOAP Fault View

 The **SOAP Fault** view is available for Web Services only. It allows you to specify whether to expect a SOAP fault and to set the SOAP fault properties. For details, see the "SOAP Fault View" on page 76.

## Attachments View

 The **Attachments** view is available for Web Services. It allows you to view the input and output attachments and their properties. For details, see the "Attachments View" on page 81.

## Security Settings View

 The **Security Settings** view, available for Web services, allows you to add tokens, message signatures, and encryption to the service calls. You can also select security scenarios and customize them. For details, see "Security Settings View" on page 82.

## Data Queries View



The **Data query** view is available when you click in the empty area of a Loop box. It allows you to add queries that appear in the Data window, and set its data navigation policies. For details, see "Navigating within the Data" on page 153.

## Data Query Properties View



The **Data Query Properties** view, only available for a REST service methods, displays the HTTP request, response and version. You can also set the headers and import the body of the request. For details, see "Data Query Properties View" on page 83.

## UFT (Unified Functional Testing)

The UFT (Unified Functional Testing) activities allow you to create tests with steps from Service Test, QuickTest Professional, or LoadRunner's Virtual User Generator.

QuickTest Professional is HP's tool for functional testing of client and Web applications. HP LoadRunner allows you to perform Load Testing on your applications and environment.

You create the tests or scripts in the original application and call them within the Service Test flow. You can also call another Service Test test from within the Test Flow.

For user interface details, see the "Toolbox Palette" on page 59

For task details, see "How to Implement UFT testing" on page 47.

## JMS Transport Overview

"The Property Sheet" on page 33 is a J2EE standard for sending messages—either text or Java objects—between Java clients. Service Test supports the sending of text messages between Java clients.

There are two scenarios for communication:

**Peer-to-Peer.** Also known as **Point-to-Point**. JMS implements point-to-point messaging by defining a message queue as the target for a message. Multiple senders send messages to a message queue, and the receiver gets the message from the queue.

**Publish-Subscribe.** Each message is sent from one publisher to many subscribers through a designated topic. The subscribers only receive messages sent after they have subscribed.

Service Test supports both types of communication by allowing you to send and receive JMS messages to and from a queue or topic. The following is a list of the built-in JMS activities.

To interpret Topic and Queue names, Service Test calls a lookup method on a JNDI context, defined in the Property Sheet's **Test Settings**. For details, see the "Test Settings View" on page 66.

For task details, see "Set up a JMS message - optional" on page 45.

### **Publish Message to Topic**

Publishes a message to multiple subscribers using a JMS topic.

Property	Description
<b>Topic name</b>	The JNDI name of the queue from which to receive the message.
<b>Message</b>	The message to publish
<b>JMS send properties</b>	JMS properties with the key/value form. You can use the key to modify JMS header or message properties.

## Receive Message from Queue

Receives a message from a JMS queue.

Property	Description
<b>Receive queue name</b>	The JNDI name of the queue from which to receive the message. If the value is empty, Service Test uses a temporary queue.
<b>JMS receive message selector</b>	An SQL expression to filter the received message. For more information on the syntax for defining selectors, see the <b>Message Properties</b> section in <a href="http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/jms/Message.html">http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/jms/Message.html</a> .  For example a key <b>JMSCorrelationID</b> , with a value of <b>4566636</b> , only receives messages whose correlation ID is 4566636 (right-click to add array elements).

## Receive Message from Topic

Receives published messages to a specific JMS topic for a subscription. Call this step after a **Subscribe to Topic** step.

Property	Description
<b>Topic name</b>	The JNDI name of the topic from which to receive the message.
<b>Subscription name</b>	The name of the subscription. Use the same subscriber name value from the previously-called <b>Subscribe to Topic</b> step.

## Send Message to Queue

Sends a message to a JMS queue.

Property	Description
<b>Send queue name</b>	The JNDI name of the queue from which to receive the message.

Property	Description
<b>Message</b>	The message to send.
<b>JMS send properties</b>	JMS properties with the key/value form. You can use the key to modify JMS header or message properties.

## Send and Receive Message From Queue

Sends a message to a specified queue and receives a message from a specified queue.

Property	Description
<b>Send queue name</b>	The JNDI name of the queue to which to send the message.
<b>Receive queue name</b>	The JNDI name of the queue from which to receive the message. If the value is empty, Service Test uses a temporary queue.
<b>Message</b>	The message to send.
<b>JMS send properties</b>	JMS properties with the key/value form. You can use the key to modify JMS header or message properties.
<b>JMS receive message selector</b>	<p>An SQL expression to filter the received message. For more information on the syntax for defining selectors, see the <b>Message Properties</b> section in <a href="http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/jms/Message.html">http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/jms/Message.html</a>.</p> <p>For example a key <b>JMSCorrelationID</b>, with a value of <b>4566636</b>, only receives messages whose correlation ID is 4566636 (right-click to add array elements).</p> <p><b>Tip:</b> To automatically generate a selector, use the <b>Automatically generate selector</b> option in the Property Sheet's <b>Test Settings</b> tab. This selector uses a correlation ID that corresponds to the request's message ID.</p>

## Subscribe to Topic

Creates a subscription for a JMS topic. Use this step before any **Receive Message from Topic** steps for the subscription

Property	Description
<b>Topic name</b>	The JNDI name of the queue from which to receive the message.
<b>Subscription name</b>	The name of the subscription. To retrieve messages on the subscription created by this step, insert a <b>Receive Message from Topic</b> step using the same Subscription Name value.
<b>JMS receive message selector</b>	An SQL expression to filter the received message. For more information on the syntax for defining selectors, see the <b>Message Properties</b> section in <a href="http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/jms/Message.html">http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/jms/Message.html</a> .



---

---

## Tasks

---

---

### How to Populate a Test

This task describes how to set up the workflow and add steps to your tests. For details on creating the test and defining user or test variables, see "Creating and Saving Tests" on page 13.

This task includes the following steps:

- "Prepare the service references - optional" on page 41
- "Create the test flow" on page 42
- "Add activities to the test" on page 42
- "Define data for the test" on page 42
- "Add additional properties - optional" on page 42
- "Create a Custom Code activity - optional" on page 43
- "Add attachments to the test - optional" on page 44
- "Set up a JMS message - optional" on page 45
- "Configure SOAP Fault information - optional" on page 45
- "Specify an event handler - optional" on page 45

#### **1 Prerequisite - open a test**

Create a new test. For details, see "How to Create a New Test" on page 15.

#### **2 Prepare the service references - optional**

For Web Service based tests, import your WSDL at this point. For details, see "How to Import a WSDL-Based Web Service" on page 116.

For REST services, build the metadata for the service. For details, see "How to Create a Template-Based REST Service" on page 125.

Skip this step when using the built-in activities, such as string manipulation.

### 3 Create the test flow

Expand the nodes of the Toolbox palette and drag **Flow Control** activities into the canvas:

- **Loop.** Drag the **Loop** activity into the canvas to add a loop other than the standard **Test Flow**. Specify a loop condition in the input properties.
- **Condition.** Drag a **Condition** activity into the Test Flow or Loop frame. Specify a condition in the input properties.
- **Sleep.** Drag a **Sleep** activity into the Test Flow or Loop frame. Indicate a time delay in milliseconds.

### 4 Add activities to the test

Expand the nodes of the Toolbox palette and drag activities into the Text Flow box within the canvas. If you added a **Condition** step, drag activities into the condition branches.

For a list of the activities, see the "Toolbox Palette" on page 59.

### 5 Define data for the test

For details, see "How to Assign Data to Test Steps" on page 161 or "How to Data Drive a Test Step" on page 165.

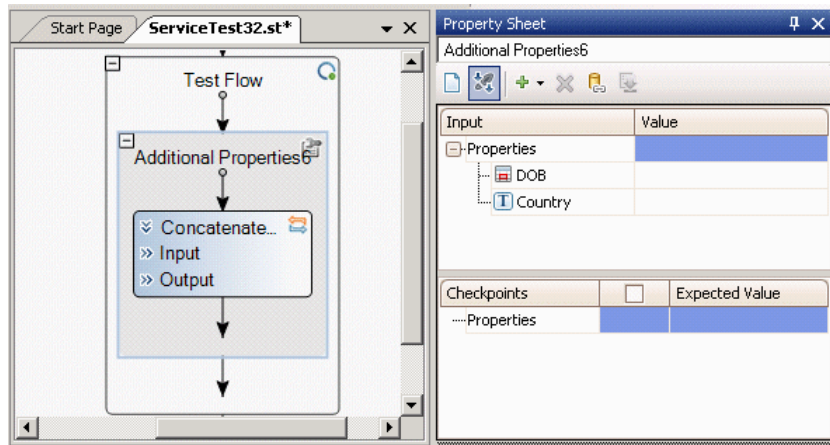
### 6 Add additional properties - optional

Use the **Additional Properties** activity to add additional input or output parameters to a built-in activity or imported service.

- a** Drag the **Additional Properties** activity from the **Miscellaneous** category, into the Test Flow.
- b** Open the **Input/Checkpoints view** in the Property Sheet.
- c** Add new input and output properties based on your needs.



- d** Drag the activity from the toolbox for which you want to utilize these new properties, into the **Additional Properties** frame within the Test Flow.



## 7 Create a Custom Code activity - optional

Select the **Custom Code** activity from the **Miscellaneous** category and drag it into a loop.



- a** Open the **Input/Checkpoints** view in the Property Sheet.
- b** Add new input and output properties based on your needs.
- c** Open the **Events** view in the Property Sheet.
- d** Double-click the **Handler** column of the **ExecuteEvent** row. Service Test opens a new tab **SharedUserCode.cs**.
- e** Locate the **TODO** section and enter your custom code. Follow the sample code in the comments and use autocompletion to write your code.
- f** Click **File > Save All** to save all buffers of the test.

For details and examples, see Appendix A, "Coding Service Test Events."

## 8 Add attachments to the test - optional

For Web Service activities that support input attachments, add an input attachment:



- a** Select the Web Service in the canvas and open the **Attachments** view in the Property Sheet.
- b** In the upper pane, select an attachment **Type: DIME** or **MIME**.
- c** Click on **Attachments** and select **Add Array Element** from the shortcut menu.



- d** Select an **Origin** for the attachment using the Browse button.
- e** Select an **Origin Type**, **Content Type**. Specify a **Content ID** or keep the default value, **Auto**.

### To validate an output attachment:



- a** Select the Web Service in the canvas and open the **Attachments** view in the Property Sheet.
- b** Click on **Attachments** in the bottom pane, in the Output Attachments node. Select **Add Array Element** from the shortcut menu.
- c** Select the check box adjacent to each item that you want to validate. Specify values for the elements being validated: **Content Type** and/or **Content ID**.



- d** To validate content, click the **Calculate the file checksum** button icon in the **Content** row. Service Test calculates the file's checksum using the MD5 Hash function.

For user interface details, see "Attachments View" on page 81. You can also check for received attachments in the test's folder, stored as files with a **bin** extension.

## 9 Set up a JMS message - optional



- a** Click in the canvas and open the **Test Settings** view in the Property Sheet. Set the Java test settings for the VM and JMS. For details, see the "Test Settings View" on page 66.
- b** Expand the **JMS** node in the toolbox and drag a JMS activity into the canvas.
- c** Set the step's properties. Click on the step in the canvas, and open the **Input/Checkpoints** view in the Property Sheet. Enter a Queue, Subscription, Topic name, and any other relevant property values.
- d** For Send activities, specify a message.
- e** For Receive activities, select the output properties you want to validate in the **Checkpoints** pane and specify their values.



For more details, see "JMS Transport Overview" on page 36.

## 10 Configure SOAP Fault information - optional

To apply negative testing to a Web service:



- a** Open the **SOAP Fault** view.
- b** Select **Fault is expected**.
- c** Provide SOAP Fault checkpoint values for the negative testing:
  - To work In the **XML** layout: Expand the SOAP nodes and define **Any** elements for the SOAP Header and Body. If relevant, provide values for **faultcode**, **faultstring**, or **faultactor**.
  - To work with **Xpath** expressions: Click the **Xpath** tab and use the Add button to add new Xpath entries. Copy the XPATH entry into the cell.

For details, see the "SOAP Fault View" on page 76.

## 11 Specify an event handler - optional

For non-custom code activities, you can define default event handlers for checkpoints, before step executions, and after step executions.

The checkpoint event handlers help you verify the output values in your test. You can use a Report, Assert, or Log function to gather information about your service.

**To add a checkpoint event handler for an activity:**

- a** Select an activity title in the canvas and open the **Events** view in the Property Sheet.
- b** In the **CodeCheckPoint** row, select **Create a default handler**.
- c** Edit the code in the **SharedUserCode.cs** tab. Locate the **TODO** section and add your custom code. Follow the sample code in the comments and use auto-complete to create an expression.
- d** To access the properties of an activity, cast it beforehand. For example:

```
ConcatenateStringsActivity cat = args.Activity as ConcatenateStringsActivity;  
args.Checkpoint.Assert.Equals(cat.Prefix+cat.Suffix, cat.Result);
```

- e** Click **File > Save All** to save all buffers of the test.

For details and examples, see Appendix A, "Coding Service Test Events."

---

**Tip:** To instruct Service Test to ignore all event handlers, select a handler and press **DELETE** to clear the field in the grid.

---

## 12 Run the test

Run the test and view the report. For more information, see "How to Run a Test" on page 187.

## How to Implement UFT testing

This task describes how to incorporate tests from other HP applications to provide Unified Functional Testing solution (UFT). These applications include QuickTest Professional, Virtual User Generator, and Service Test. For details, see "UFT (Unified Functional Testing)" on page 36.

This task includes the following steps:

- "Prerequisites" on page 47
- "Add a QuickTest test activity - optional" on page 48
- "Add a LoadRunner script activity - optional" on page 49
- "Add a Service Test test - optional" on page 49

### 1 Prerequisites

Make sure you have installed the application whose test/script you want to call:

- QuickTest test step: Make sure you have QuickTest Professional 11.00. Create the test in QuickTest Professional and run it at least once.
- Virtual User Generator step: Make sure you have HP LoadRunner 11.00. Generate a test with the Virtual User Generator, or use a test that you enabled for load testing in Service Test.

## 2 Add a QuickTest test activity - optional

- a** In Service Test, expand the **Unified Functional Testing** node in the Toolbox palette. Drag the **Call QuickTest Professional Test** activity into the canvas.
- 

**Tip:** Do not insert a call to a QuickTest test that contains a call to a Service Test test, as this can cause unexpected behavior.

---

- b** Open the Property Sheet's **Input/Checkpoints** view, and click the **Select QTP Test** button. Select a QuickTest Professional version 11.00 (or higher) test.
  - c** If the QTP test changed at its source, you can update it in Service Test. In the **Input/Checkpoints** view, click **Update properties**.
  - d** Service Test imports the properties defined in the QuickTest test. You can edit the properties manually, data drive them, or load values from a replay.
  - e** Add other relevant steps to your test.
  - f** Save and run the test.
- 

**Note:** The first time you invoke QuickTest Professional in a test step, you may need to wait several seconds for Service Test to invoke QTP and load the test.



---



### 3 Add a LoadRunner script activity - optional

- a In Service Test, expand the **Unified Functional Testing** node in the Toolbox palette. Drag the **Call Virtual User Generator Script** activity into the canvas.
- b Open the Property Sheet's **General** view, and click the Browse button in the **Script Path** row. Select a VuGen (Virtual User Generator) .usr file.
- c Add other relevant steps to your test.
- d Save and run the test.

### 4 Add a Service Test test - optional

- a Expand the **Unified Functional Testing** node in the Toolbox palette. Drag the **Call Service Test test** activity into the canvas.
-  b Open the Property Sheet's **General** view., and click the **Select ST Test** button. Select a test version 11.00 (or higher).
- c To update a test that has already been loaded, click **Update ST Test**.
-  d To specify a custom directory for the results, click the Browse button in the **Results Directory** row in the **General** view. Select a directory for the results.
- e Add other relevant steps to your test.
- f Save and run the test.

## How to Create a Call to a Java Class

The Call Java Class activity lets you to add Java steps to your test script. This feature allows you to incorporate existing Java code into your test.

This task describes how to create the Java call and includes the following steps:

- "Prerequisite - open a test" on page 50
- "Set the global Java settings - optional" on page 50
- "Implement the Service Test Java interface" on page 50

- ▶ "Package your custom step - optional" on page 51
- ▶ "Add a Call Java Class activity" on page 51
- ▶ "Open the Java Class Setting dialog box" on page 51
- ▶ "Set the Java Call properties" on page 51
- ▶ "Provide Input parameters" on page 51
- ▶ "Run the test." on page 52

## 1 Prerequisite - open a test

Create a new test. For details, see "How to Create a New Test" on page 15.

## 2 Set the global Java settings - optional



To set global VM (Virtual Machine) settings, click in the canvas outside of the Test Flow, and open the **Test Settings** view in the Property Sheet. For details, see the "Test Settings View" on page 66.

## 3 Implement the Service Test Java interface

Run javac.exe in JDK 1.6 or earlier, to implement the Service Test interface. The interface, **ServiceTestCall.java**, is located in the `<ST_INSTALL_HOME>\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java` folder.

This interface provides the essential information to Service Test: input properties, output properties, and a point of entry.

For example, the following code implements the "Hello World" sample provided with Service Test:

```
javac "C:\Program Files\HP\Service Test 11\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java\*.java" "C:\Program Files\HP\Service Test 11\addins\ServiceTest\JavaCall\Java Interface\src\hp\st\ext\java\sample\*.java"
```

#### 4 Package your custom step - optional

Package your java classes into a jar file. This is optional, since you can also provide a package root for the class.

#### 5 Add a Call Java Class activity

In Service Test, expand the **Java** node in the Toolbox palette, and drag the **Call Java Class** activity into the canvas.

#### 6 Open the Java Class Setting dialog box



Click on the Java step in the canvas, and open the **Input/Checkpoints view** in the Property Sheet. Click the **Select Java File** button to open the Call Java Class Settings dialog box.

For general guidelines, see "Java Testing" on page 32.

#### 7 Set the Java Call properties

In the Call Java Class Settings dialog box:

- a** Provide a classpath. If you packaged your Java step, click the **Jar** button and point to the jar file. Alternatively, click the **Package root** button to select a package root folder.
- b** Click the Browse button in the **Selected Java Class** box, to locate the class within the jar file or the folder. Make sure it is a class that implements the **ServiceTestCall** interface.
- c** To provide additional classpaths, click the **Jar** or **Folder** buttons in the **Additional Classpaths** section to browse for a jar file or classpath folder. Click **Add** to move the contents into the list.
- d** Click **OK** to save the Java Call settings.

For user interface details, see the "Call Java Class Settings Dialog Box" on page 95.

#### 8 Provide Input parameters



In the Property Sheet's **Parameter** view, provide values for the step's input parameters.

**9 Run the test.**

Run the test and view the results in the Run Results Viewer.

---



---

## References

---



---

### The Service Test User Interface

The Service Test interface lets you design a test by dragging activities into a canvas.

This section includes:










- [Service Test Main Window](#) on page 53
- [Tests Pane](#) on page 57
- [Toolbox Palette](#) on page 59
- [Property Sheet](#) on page 63
- [Output Tab](#) on page 93
- [Errors Tab](#) on page 94
- [Call Java Class Settings Dialog Box](#) on page 95




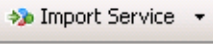
### Service Test Main Window

This screen is the main window box of the Service Test.

<b>To access</b>	Select <b>Start &gt; HP Service Test 11.00 &gt; HP Service Test 11.00</b>
<b>Relevant tasks</b>	"How to Populate a Test" on page 41

User interface elements are described below (unlabeled elements are indicated by angle brackets):

UI Elements (A-Z)	Description
	<b>Create New Test.</b> Opens the Create New Test dialog box.
	<b>Open Test.</b> Prompts you to open an existing test.
	<b>Add New Test.</b> Opens the Add New Test dialog box.
	<b>Save.</b> Saves the file in the active tab. <b>Tip:</b> Select <b>File &gt; Save Test As...</b> to save the test to a new location.
	<b>Save All.</b> Saves all open files.
	<b>ALM/QC Connection.</b> Opens the HP ALM/QC Connection dialog box for defining server information and logging into an ALM/QC project.
	<b>Enable Load Testing.</b> Enables the test for Load Testing. <b>Note:</b> <ul style="list-style-type: none"> <li>➤ This button adds a menu to the <b>Run</b> button, to run the test in Load Testing mode.</li> <li>➤ The button is disabled if you created the test with the Load Test template.</li> </ul>
	<ul style="list-style-type: none"> <li>➤ <b>Run Project.</b> Saves, compiles and runs the current project.</li> <li>➤ <b>Run Test (Load Test).</b> Saves, compiles and runs the current test in Load Test mode with the <b>mdrv</b> driver.</li> </ul> <b>Note:</b> The <b>Run Test</b> option is only available for Load Test type tests (created with the Load Test template or converted from the standard template to a load test).
	<ul style="list-style-type: none"> <li>➤ <b>Break Test.</b> Pauses the test run.</li> <li>➤ <b>Continue Test.</b> Resumes the test run from the point of the Break.</li> </ul>

UI Elements (A-Z)	Description
	<p><b>Stop.</b> Stops the current test run.</p>
	<p>Zoom Controls:</p> <ul style="list-style-type: none"> <li>▶ <b>Default Zoom.</b> Restores the default enlargement of the objects in the canvas.</li> <li>▶ <b>Zoom in.</b> Enlarges the objects in the test canvas.</li> <li>▶ <b>Zoom out.</b> Reduces the objects in the test canvas.</li> </ul>
	<p>The following step commands are only active when the test is in Break mode.</p> <ul style="list-style-type: none"> <li>▶ <b>Step Over.</b> Executes the current statement. If the statement calls another procedure, Service Test will not step into that procedure. It will run the procedure, and advance to the next statement in the current procedure.</li> <li>▶ <b>Step Into.</b> Executes the current statement. If the statement calls another procedure, Service Test steps into that procedure.</li> <li>▶ <b>Step Out.</b> If you used <b>Step Into</b> and the control moved to a called procedure, <b>Step Out</b> runs the current procedure, and returns to the procedure from which it was called.</li> </ul>
	<p>Opens one of the Import dialog boxes:</p> <ul style="list-style-type: none"> <li>▶ <b>Import Service from URL/UDDI.</b> Opens the Import Service from URL/UDDI dialog box. For user interface details, see the "Import Service from URL/UDDI Dialog Box" on page 134.</li> <li>▶ <b>Import Service from File/ALM Application Component.</b> Opens the Select Service dialog box for importing a WSDL from the file system or HP ALM/QC with HP Service Test Management. For user interface details, see the "Select Service Dialog Box" on page 133.</li> </ul>

UI Elements (A-Z)	Description
<.Editor / Designer pane>	<p>A series of tabs, to view and edit the file selected in the left pane: For example:</p> <ul style="list-style-type: none"> <li>➤ visual display of the test (default).</li> <li>➤ C Sharp files (.cs)</li> <li>➤ XSD files (.xsd)</li> <li>➤ Solution file (.sln)</li> </ul>
<b>Data Window tab</b>	<p>Displays the data from an Excel or XML file, or local tables. You can use this data in your tests.</p> <ul style="list-style-type: none"> <li>➤ <b>Data Explorer.</b> An expandable/collapsible tree hierarchy of the table data.</li> <li>➤ <b>New.</b> Creates a new data node in the Data Explorer through: <ul style="list-style-type: none"> <li>➤ <b>Excel File.</b> Opens the <b>Add New Excel File Data Query</b> dialog box for creating a new set of data. For details, see the "Add New Excel File Data Query Dialog Box" on page 168.</li> <li>➤ <b>Local Table.</b> Opens the <b>Add New Local Table Data Query</b> wizard. For details, see the "Add New Local Table Data Query Dialog Box" on page 169.</li> <li>➤ <b>XML.</b> Opens the <b>Add New XML Data Query</b> dialog box. For details, see the "Add New XML Data Query Dialog Box" on page 170.</li> </ul> </li> <li>➤ <b>Delete.</b> Removes the selected data query.</li> </ul>
<b>Errors tab</b>	<p>Displays errors, warning, and messages about the test run.</p> <p>The default location is the bottom pane.</p>
<b>Output tab</b>	<p>Displays the Output log for the compilation and test run. For details, see "Output Tab" on page 93.</p> <p>The default location is the bottom pane.</p>
<b>Tests pane</b>	<p>Displays the test components in a tree hierarchy. For details, see the "Tests Pane" on page 57.</p> <p>The default location is the left pane.</p>






UI Elements (A-Z)	Description
<b>Property sheet</b>	A series of screens showing the general properties, schemas, snapshots, and event information for the activity selected in the canvas. For details, see "Property Sheet" on page 63. <b>Note:</b> Only active when viewing the canvas.
<b>Toolbox palette</b>	A tree view of all of the available activities. This includes built-in activities and operations of imported services. For details, see the "Toolbox Palette" on page 59. The default location is the left pane.

## Tests Pane

The **Tests** pane enables you to view all the test components in a tree hierarchy.

<b>To Access</b>	<b>View &gt; Tests</b> (Ctrl+Alt+L)
<b>Relevant Tasks</b>	"How to Create a New Test" on page 15
<b>See Also</b>	"Test User Interface" on page 19

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

UI Elements	Description
	<b>Show All.</b> Shows/hides all files associated with current test.
	<b>Refresh.</b> Refreshes the Tests tree.
	<b>Open.</b> Opens a tab with the contents of the selected file. <b>Note:</b> Only available for the st, canvas file.

UI Elements	Description
<Test> Tree	<p>Tree hierarchy of the test components. Expanding the parent node displays all the references and dependent files.</p> <p><b>Tip.</b> To view all files, click the <b>Show All</b> button.</p>
<Test> Tree - Solution context menu	<ul style="list-style-type: none"> <li>➤ <b>Build Solution.</b> Begins a new build (F8).</li> <li>➤ <b>Rebuild Solution.</b> Rebuilds the current solution (Alt+F8).</li> <li>➤ <b>Clean Solution.</b> Performs a clean build.</li> <li>➤ <b>Add.</b> Allows you to add a New Test, Existing Test, New Solution Folder, or another Item to the solution tree.</li> <li>➤ <b>Paste.</b> Pastes a solution from the clipboard.</li> <li>➤ <b>Rename.</b> Renames the current solution node.</li> </ul>
<Test> Tree - Project context menu	<ul style="list-style-type: none"> <li>➤ Build commands: <b>Build, Rebuild, Clean</b></li> <li>➤ References: <b>Add Reference</b></li> <li>➤ Open/Run: <b>Open With..., Run Project</b></li> <li>➤ Edit: <b>Cut, Paste, Remove, Rename</b></li> </ul>
<Test> Tree - References context menu	<ul style="list-style-type: none"> <li>➤ <b>Add Reference.</b> Opens the Add Reference dialog box.</li> </ul>
<Test> Tree - File context menu	<ul style="list-style-type: none"> <li>➤ <b>Refresh</b></li> <li>➤ <b>Remove</b></li> </ul>
<Test> Tree - Test file (*.st) context menu	<ul style="list-style-type: none"> <li>➤ Open commands: <b>Open, Open With..., Open Containing Folder in Explorer</b></li> <li>➤ Add commands: <b>New Item, Existing Item, New Folder, Existing Folder, New Dependent Item, Existing Dependent item</b></li> <li>➤ Edit commands: <b>Cut, Copy, Paste, Delete, Rename</b></li> </ul>






## Toolbox Palette


The Toolbox palette enables you to view all of the available activities. This includes imported services and built-in activities such as that you can use in your visual design, such as **Replace String** or **File Exists**.

<b>To Access</b>	Use one of the following: <ul style="list-style-type: none"> <li>▶ Click the <b>Toolbox</b> tab in the left pane (<b>Plain</b> layout)</li> <li>▶ <b>View &gt; Toolbox</b></li> </ul>
<b>Important Information</b>	<ul style="list-style-type: none"> <li>▶ Drag components from this tab into the canvas.</li> <li>▶ When you drag an activity into the canvas, it becomes a step in the test.</li> <li>▶ Only available when viewing the <b>Toolbox</b> tab.</li> </ul>
<b>Relevant Tasks</b>	"Create the test flow" on page 42

## Toolbar Controls

The following toolbar controls elements are included in the Toolbox Palette:

Tree Elements	Description
	<b>Expand All.</b> Expands all nodes in the Activity list.
	<b>Collapse All.</b> Collapses all node in the Activity list.
	<b>Add Resource.</b> Adds a resource under the selected node. <b>Note:</b> Located adjacent to the Filter box, for the REST Services node only.
	<b>Add Method.</b> Adds a method under the selected node. <b>Note:</b> Located adjacent to the Filter box, for the REST Services node only.
	<b>Delete.</b> Delete a service, resource, or method. <b>Note:</b> Located adjacent to the Filter box, for the REST Services node only.

Tree Elements	Description
	<b>Rename.</b> Rename a service, resource, or method. <b>Note:</b> Located adjacent to the Filter box, for the REST Services node only.
<Filter box>	Filters the toolbox display by the entered text.

## Activity Groups

The following Activity groups are included in the Toolbox palette (in alphabetical order):


Group name	Description
<b>File System</b>	Provides access to the following file related activities: <ul style="list-style-type: none"> <li>➤ <b>File Exists.</b> Returns true if the file exists.</li> <li>➤ <b>Folder Exists.</b> Returns true if the folder exists.</li> <li>➤ <b>File Copy.</b> Copies a file to a new destination.</li> <li>➤ <b>File Move.</b> Moves a file to a new destination.</li> <li>➤ <b>File Create.</b> Creates a new file.</li> <li>➤ <b>File Delete.</b> Deletes a file.</li> <li>➤ <b>File Rename.</b> Renames a file.</li> <li>➤ <b>Write to File.</b> Writes to an existing or new file.</li> <li>➤ <b>Read from File.</b> Retrieves text from a file.</li> </ul>
<b>Flow Control</b>	Provides access to the following control flow activities: <ul style="list-style-type: none"> <li>➤ <b>Condition.</b> Allows you use a branching mechanism based on the value of a property.</li> <li>➤ <b>Loop.</b> A loop frame whose properties can be set independent of the Test Flow.</li> <li>➤ <b>Break.</b> Halts the test execution.</li> <li>➤ <b>Continue.</b> Resumes the test execution after a break.</li> <li>➤ <b>Sleep.</b> Pauses the test execution for a specific amount of time.</li> </ul> <p>For details, see "Flow Control" on page 25.</p>
<b>Java</b>	Calls a Java class. For details see the "Call Java Class Settings Dialog Box" on page 95.

Group name	Description
<b>JMS</b>	<p>Activities that allow you to handle JMS messages:</p> <ul style="list-style-type: none"> <li>▶ Publish Message to Topic</li> <li>▶ Receive Message from Queue</li> <li>▶ Receive Message from Topic</li> <li>▶ Send Message to Queue</li> <li>▶ Send and Receive Message from Queue</li> <li>▶ Subscribe to Topic</li> </ul> <p>For details, see "JMS Settings" on page 66.</p>
<b>Load Testing</b>	<p>Activities that allow you prepare the test for Load Testing, such as <b>Start Transaction</b> and <b>End Transaction</b>.</p>
<b>Miscellaneous</b>	<p>Provides access to the following miscellaneous activities:</p> <ul style="list-style-type: none"> <li>▶ <b>Custom Code</b>. Allows you to program a step to execute your own activity.</li> <li>▶ <b>Additional Properties</b>. Defines additional properties for built-in or imported services.</li> <li>▶ <b>Set Test Variable</b>. Defines global variables for your test.</li> <li>▶ <b>Run Program</b>. Invokes an application the Service Test machine.</li> <li>▶ <b>End Program</b>. Closes an open application.</li> <li>▶ <b>Report Message</b>. Sends a custom message to the Run Results.</li> </ul>
<b>Network</b>	<p>Activities that allow you to handle the transport method, such as <b>HTTP</b>.</p>
<b>Rest Services</b>	<p>Lets you define resources and methods for a REST Service. For details, see "How to Create a Simple Test for a REST Service" on page 122.</p>




Group name	Description
<b>String Manipulation</b>	<p>Provides access to the following string manipulation activities:</p> <ul style="list-style-type: none"> <li>▶ <b>Concatenate Strings.</b> Joins two strings.</li> <li>▶ <b>Replace String.</b> Replaces part of a string with an alternate string.</li> <li>▶ <b>Substring.</b> Extracts the characters between two specified locations of a string.</li> <li>▶ <b>Trim String.</b> Removes whitespace characters from the beginning or end of a string.</li> </ul>
<b>System</b>	<p>Provides access to the operating system variables:</p> <ul style="list-style-type: none"> <li>▶ Set OS Environment Variable</li> <li>▶ Get OS Environment Variable</li> </ul>
<b>Unified Functional Testing</b>	<ul style="list-style-type: none"> <li>▶ <b>Call Service Test test.</b> Invokes a Service Test test when the current Test Flow or loop reaches this step.</li> <li>▶ <b>Call QuickTest Professional test.</b> Invokes a QTP (QuickTest Professional) test when the current Test Flow or loop reaches this step. Do not insert a call to a QuickTest test that contains a call to a Service Test test, as this can cause unexpected behavior.</li> <li>▶ <b>Call VuGen script.</b> Runs a VuGen (LoadRunner's Virtual User Generator) script.</li> </ul> <p><b>Note:</b> The first time you invoke QTP in a test step, you may need to wait several seconds for Service Test to invoke QTP and load the test.</p> <p>For details, see "How to Implement UFT testing" on page 47.</p>
<b>Web Services</b>	<ul style="list-style-type: none"> <li>▶ <b>SOAP Request.</b> Sends a manual SOAP request to the server, usually in XML format. You can import a schema and load XML for this activity. For details, see "Input/Checkpoints View" on page 70.</li> <li>▶ <b>&lt;Imported Web service methods&gt;.</b> For imported Web services, a tree hierarchy showing: <ul style="list-style-type: none"> <li>▶ level 1 - services</li> <li>▶ level 2 - ports</li> <li>▶ level 3 - operations</li> </ul> </li> </ul>








## Property Sheet

Displays a series of tabs showing the general properties, schemas, and event information for the activity selected in the canvas.





<b>To Access</b>	<b>View &gt; Property Sheet</b>
<b>Important Information</b>	<ul style="list-style-type: none"> <li>▶ Only active when the canvas is visible.</li> <li>▶ The toolbar displays different buttons based on the type of activity or view.</li> </ul>  <ul style="list-style-type: none"> <li>▶ The buttons to the left of the horizontal divider, are <b>View</b> buttons, such as <b>General</b> or <b>Events</b> views.</li> <li>▶ The buttons to the right of the horizontal divider, are <b>Action</b> buttons, such as <b>Data-Drive</b> and <b>Load XML</b>. These differ based on the view. For details, see the specific view.</li> </ul>
<b>Relevant Tasks</b>	<ul style="list-style-type: none"> <li>▶ "How to Populate a Test" on page 41</li> <li>▶ "How to Set Security for a Web Service" on page 118</li> </ul>

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

UI Elements	Description
	<b>Test Settings view.</b> Java related settings that apply to all test steps. This is only visible when clicking in the canvas, outside of the Test Flow. For details, see "Test Settings View" on page 66.
	<b>Service Test test view.</b> The path and properties of the Service Test test to be called from this step. This is only visible when selecting a <b>Call Service Test test</b> activity.
	<b>General view.</b> Shows the general properties of the selected element. By default it shows the <b>Step ID</b> as it appears in the Toolbox, and the <b>Name</b> as it is shown in the canvas.  For details, see "General View" on page 68.

UI Elements	Description
	<p><b>Input/Checkpoints view.</b> The input properties and the checkpoint values for those properties. The top section shows the input properties; the bottom shows the checkpoints.</p> <p><b>Tip:</b> A triangular icon indicates an optional element. Toggle the triangle to include or exclude an element. A filled-in triangle indicates an included element.</p> <p><b>For REST and Custom Code:</b> Use the  Add button to create new parameters.</p> <p>For details, see "Input/Checkpoints View" on page 70. For <b>REST</b> and <b>HTTP</b> types, see "REST HTTP Input View" on page 78.</p>
	<p><b>SOAP Fault view.</b> The SOAP fault settings for the Web Service steps.</p>
	<p><b>HTTP view.</b> The body of the input request and the checkpoints (HTTP activities and REST services only).</p> <p>For details, see the "HTTP View" on page 78.</p>
	<p><b>REST HTTP Input view.</b> The HTTP input elements for REST services.</p> <p>For details, see the "REST HTTP Input View" on page 78.</p>
	<p><b>Events view.</b> Lets you select a handler function for a step's events, or to open an editor to write code describing what to do when the event occurs.</p> <p><b>Tip:</b> The <b>Create a default handler</b> option lets you create a default event handler. For details, see the "Events View" on page 80.</p>
	<p><b>Attachments view.</b> A list of attachments associated with the Web Service step. For details, see "Attachments View" on page 81.</p> <p><b>Note:</b> This view is only available for Web Service steps.</p>



UI Elements	Description
	<p><b>Security Settings view.</b> Allows the setting of Security Scenarios and security tokens for the Web Service. For details, see "Security Settings View" on page 82.</p> <p><b>Note:</b> This view is only available for Web Service steps.</p>
	<p><b>Test Variables view.</b> Displays the User and Test variables, and allows you to create Test Variable profiles. For details, see "Test Variables View" on page 77.</p> <p><b>Note:</b> This view is only available when selecting the <b>Start</b> or <b>End</b> steps, or by clicking in the canvas outside of the Test Flow.</p>
	<p><b>Data Queries view.</b> Allows you to select data queries for the loop, and set their policies. For details, see the "Data Queries View" on page 82.</p> <p><b>Note:</b> This view is available when selecting the Test Flow frame or a loop's frame in the canvas.</p>
	<p><b>Data Query Properties view.</b> Allows you to define and edit data relations between tables. For details, see the "Define/Edit New Data Relation Dialog Box" on page 181.</p> <p><b>Note:</b> This view is only available when selecting a query in the Data window.</p>
<p>&lt;data grid&gt;</p>	<p>Information about the step selected in the canvas, depending on the view:</p> <ul style="list-style-type: none"> <li>▶ <b>General view.</b> Activity name and comments.</li> <li>▶ <b>Input/Checkpoints view.</b> Properties, checkpoints, and their values,</li> <li>▶ <b>Events view.</b> A list of the events and their handlers.</li> <li>▶ <b>Data Query view.</b> A list of queries for the selected step.</li> </ul>
<p><b>Comments</b></p>	<p>An editable note describing the purpose of the step. Service Test also places the comment in the test report.</p>

## Test Settings View

Test settings apply to all the activities in the test. User interface elements are described below:

### Load Settings

UI Elements	Description
<b>Load enabled (read-only)</b>	Indicates whether the test is enabled for load testing. To enable load testing, select <b>Test &gt; Enable Test for Load</b> . <b>Note:</b> Once a test is enabled for load testing, you cannot disable it.

### JVM (Java Virtual Machine) Settings

UI Elements	Description
<b>Classpath</b>	The vendor implementation of JMS classes together with any other required supporting classes, as determined by the JMS implementation vendor.
<b>Additional VM Parameters</b>	Extra parameters to send to the JVM such as Xbootclasspath, and any parameters specified by the JVM documentation.

### JMS Settings

UI Elements	Description
<b>JNDI initial context factory</b>	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.
<b>JNDI provider URL</b>	The URL of the service provider. For example: Websphere - iiop://myserver:myport
<b>JMS connection factory</b>	The JNDI name of the JMS connection factory. This setting is unique per test.
<b>JMS security principal</b>	Identity of the principal (for example the user) for the authentication scheme.

UI Elements	Description
<b>JMS security credentials</b>	The principal's credentials for the authentication scheme.
<b>Number of JMS connections per process</b>	The number of JMS connections each execution process creates. The default is 1, and the maximum is 50. The fewer connections you have, the better your performance.
<b>Received message timeout options</b>	<p>The timeout for received messages. The default is <b>No wait</b>.</p> <ul style="list-style-type: none"> <li>▶ <b>Indefinite wait.</b> Wait as long as required for the message before continuing.</li> <li>▶ <b>No wait.</b> Do not wait for the Receive message, and return control to the script immediately. If there was no message in the queue, the operation fails.</li> <li>▶ <b>User defined timeout.</b> Wait a specified number of seconds for the message. If it does not arrive, the operation fails. (default)</li> </ul>
<b>User defined timeout</b>	The amount of seconds to wait for the message before timing out. The default is 20 seconds.
<b>Automatically generate selector</b>	<p>Generates a selector for the response message with the correlation ID of the request (<b>No</b> by default). Each JMS message sent to the server has a specific ID. Enable this option if you want Service Test to automatically create a selector that includes the message ID.</p> <p><b>Note:</b> This option only affects the <b>Send and Receive Message from Queue</b> activity.</p>

## General View




User interface elements are described below (unlabeled elements are shown in angle brackets). The properties differ based on the activity type.

UI Elements (A-Z)	Description
<b>Properties - default</b>	<ul style="list-style-type: none"> <li>➤ <b>Step ID.</b> A unique ID for the step.</li> <li>➤ <b>Name.</b> Step name as it should appear in the canvas.</li> </ul>
<b>Properties - HTTP</b>	<ul style="list-style-type: none"> <li>➤ <b>Proxy.</b> The settings for the proxy server hosting the service contract: <b>Server</b> (URL and port when required), <b>Username</b>, and <b>Password</b>.</li> <li>➤ <b>Authentication.</b> The user credentials settings for obtaining a service contract: <b>Username</b> and <b>Password</b>.</li> <li>➤ <b>Connection Type:</b> Keep-Alive (default) or Close.</li> <li>➤ <b>Timeout.</b> The HTTP timeout in milliseconds.</li> <li>➤ <b>ClientCertificate.</b> Details about the client certificate: <ul style="list-style-type: none"> <li>➤ <b>File System:</b> The full path of the certificate file.</li> <li>➤ <b>Windows Store:</b> The certificate properties <b>StoreLocation</b>, <b>StoreName</b>, <b>X509Type</b>, and <b>X509FindValue</b>.</li> <li>➤ <b>Password.</b> The certificate's password.</li> </ul> </li> <li>➤ <b>Maximum Automatic Retries.</b> The number of times to attempt accessing a page, by default 3.</li> <li>➤ <b>Allow Redirections.</b> Allow the step to redirect to another URL. Default is true.</li> <li>➤ <b>Reuse Cookies.</b> Enables the reusing of cookies for the current step. Default is false.</li> </ul>
<b>Properties - Service Test Test/ QTP Test / VuGen Script</b>	<ul style="list-style-type: none"> <li>➤ <b>Script Path.</b> The path of the Service Test or QTP test, or VuGen (LoadRunner's Virtual User Generator) script to run.</li> </ul>

UI Elements (A-Z)	Description
<b>Properties - Web Services</b>	<p>Service Test derives these properties from the WSDL file:</p> <ul style="list-style-type: none"> <li>▶ <b>Endpoint Address.</b> The endpoint location of the service as derived from the WSDL.</li> <li>▶ <b>SoapAction.</b> An expression indicating the SOAP action, for example, HP.SOAQ.SampleApp/IHPFlights_Service/DeleteFlightOrder.</li> <li>▶ <b>ContentType.</b> For example text/xml; charset=utf-8.</li> <li>▶ <b>IsOneWay.</b> A variable indicating whether the service is a one way service: true or false.</li> </ul>
<b>Comment</b>	<p>An editable note describing the purpose of the step. Service Test also places the comment in the test report.</p>

## Input/Checkpoints View

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


UI Elements (A-Z)	Description
<properties list>	<p>A list of all the properties. The fields differ per step type:</p> <ul style="list-style-type: none"> <li>▶ For the Start step: <b>Test Input Parameter</b></li> <li>▶ For the End step: <b>Test Output Parameter</b></li> <li>▶ For activities and Web Services: <b>Input</b> properties</li> <li>▶ For SOAP Requests (under Web Services category): <ul style="list-style-type: none"> <li>▶ <b>Standard Schema.</b> A schema for the SOAP request. <ul style="list-style-type: none"> <li>▶  <b>Import Schema</b> . Imports a schema XSD file.</li> <li>▶  <b>Load XML</b> . Loads XML with values for the schema.</li> <li>▶  . Deletes the current schema.</li> </ul> </li> </ul> </li> <li>▶ For HTTP activities: <ul style="list-style-type: none"> <li>▶ <b>URL.</b> The endpoint URL of the service.</li> <li>▶ <b>HTTP method.</b> GET, POST, PUT, DELETE, TRACE, OPTIONS, HEAD, and CONNECT.</li> <li>▶ <b>HTTP version.</b> 1.0 or 1.1</li> <li>▶ <b>Request Headers.</b> Keys and their values, and arrays.</li> <li>▶ <b>Checkpoints.</b> HTTP version, Status code, Status description, and Response body.</li> </ul> </li> <li>▶ For JMS activities, see "JMS Transport Overview" on page 36.</li> </ul>
<loop settings>	<p>The loop settings for the iteration such as the number of iterations, conditions, and so forth.</p> <p><b>Note:</b> To set these properties, click on the frame of the <b>Test Flow</b> or <b>Loop</b>.</p> <p>For details, see the "Loop Settings" on page 72.</p>

UI Elements (A-Z)	Description
<checkpoint list>	<p>A list of the step's output parameters, displaying the following columns:</p> <ul style="list-style-type: none"> <li>▶ <b>Checkpoints.</b> A list of the output parameters.</li> <li>▶ <b>Validate.</b> When checked, validates the current output parameters when running the step. When unchecked, ignores the parameters.</li> <li>▶ <b>Value.</b> The expected value for the output parameter. Provides a dropdown list for comparison operators, number scrolling, and boolean values. You can manually enter the expected value into the cell.</li> </ul>
<context menu>	<p>Provides shortcuts for including properties and setting their values:</p> <ul style="list-style-type: none"> <li>▶ <b>Set Auto-value.</b> Inserts a sample value for the argument, based on its data type.</li> <li>▶ <b>Include All.</b> Includes all properties.</li> <li>▶ <b>Exclude All.</b> Excludes all properties.</li> <li>▶ <b>Copy XPath.</b> Copies the XPath expression of the value to the clipboard.</li> <li>▶ <b>Link to a data source.</b> Opens the Select Link Source dialog box.</li> <li>▶ <b>Clear cell contents.</b> Clears the contents of the cell.</li> </ul> <p><b>Tip:</b> Right-click in <b>Values</b> column to see the menu.</p>
<b>Checkpoint properties</b>	<p>Additional checkpoint properties:</p> <ul style="list-style-type: none"> <li>▶ <b>Trim whitespace (from start and end of string).</b> Removes whitespace before and after the text.</li> <li>▶ <b>Ignore case.</b> When looking for a match, ignores the case.</li> <li>▶ <b>Stop test if checkpoint fails.</b> Exits the test run if the checkpoint fails. You set this individually for each checkpoint.</li> </ul> <p><b>Note:</b> You must click on a row in the Checkpoint section to see these properties. The first two properties are only available for string data types.</p>




UI Elements (A-Z)	Description
<b>Send request to service.</b>	<p>Sends the step's request to the service when running the test. This is the normal behavior for Web Services and most types of actions. It is activated by default.</p> <p>To send messages using JMS transport, clear this check box. Link a subsequent JMS step to the output of this step, to obtain the value of a JMS key.</p>
<b>Value column</b>	<p>The property value as a constant value or a query.</p> <p>An icon adjacent to the value provides information about the property, such as read-only, data type, optional, and allows you to choose values when appropriate.</p> <p>For a list of the most common icons, see the "Value Column Icons" on page 73.</p> <p><b>Tip:</b> Click inside a row to display the relevant icons.</p> <p><b>Note:</b> To view the buttons and icons, click within a row of the <b>Value</b> column.</p>

## Loop Settings

The following iteration settings are available for the Test Flow or other custom loops:





UI Elements (A-Z)	Description
<b>For Loop</b>	Performs the Test Flow or custom loop the number of times that you specify in the <b>Number of Iterations</b> box.
<b>Do While Loop - Use condition</b>	<p>Repeats the loop as long as the specified condition is met:</p> <ul style="list-style-type: none"> <li>▶ <b>Variable.</b> The variable to evaluate. Use the Select Link Source button  to enter a query expression.</li> <li>▶ <b>Operator.</b> A comparison operator, relevant to the variable such as =, !=, <b>Contains</b>, <b>Starts</b> and <b>Regex</b>.</li> <li>▶ <b>Value.</b> The value with which to compare the result.</li> </ul>









UI Elements (A-Z)	Description
<b>Do While Loop - Use event</b>	<p>Performs iterations until the event returns True. This mode is useful for complex conditions that can be defined in an event handler. When you select this option, the Input Properties grid opens.</p> <ul style="list-style-type: none"> <li>▶ Click  . to define properties.</li> <li>▶ Click  . to open the <b>Events</b> view. Click <b>Create a default handler</b> in the <b>Handler</b> column.</li> <li>▶ Modify or add code to the event handler method that defines your condition.</li> </ul>
<b>For Each Loop</b>	<p>Performs an iteration for each element in the associated array or collection of objects. Data is selected using the Select Link Source button  .</p> <p><b>Note:</b> When you delete data from a data table, Service Test continues to run an iteration for that row, with empty values. To remove an entire row of a data, click the row and select <b>Delete</b> from the shortcut menu (only with Excel installations).</p>

## Value Column Icons



The following icons represent information, dropdown lists, or actions within the Parameter view's **Value** column:




UI Elements	Description
	<b>Include argument in the request.</b> Toggle to clear the triangle and exclude the argument.
	<b>Include an array element.</b> Toggle to clear the box and exclude the array element.
	<b>Set to NIL.</b> Toggle to clear the icon and remove the NIL value assignment.
	<b>Read-only node.</b> Values that cannot be changed such as properties linked to a data source other than a constant value, or nodes with read-only attribute in the activity signature, first HTTP header, and so forth.

UI Elements	Description
	<b>Warning.</b> A warning related to the data query. For example, <b>The data types of the link source and link destination do not match.</b>
	<ul style="list-style-type: none"> <li>▶ For <b>Input properties:</b> A dropdown list of possible values. For example, for boolean values it provides a list of values for boolean type data: <b>true</b>, <b>false</b>, <b>0</b>, or <b>1</b>. For date types, it opens a calendar. This dropdown arrow is located adjacent to the <b>Link to a data source</b> button .</li> <li>▶ For <b>Checkpoints:</b> A dropdown list of the relevant comparison operators such as: <b>=</b>, <b>!=</b>, <b>&gt;</b>, <b>&gt;=</b>, <b>&lt;</b>, <b>&lt;=</b>, <b>Starts</b>, <b>Ends</b>, <b>Contains</b>, or <b>Regex</b>. The greater than operator, <b>&gt;</b>, when used with strings, indicates that it appears later in the alphabet.</li> </ul>
	<b>Scroll.</b> A scroll control for integer data types.
	<b>Browse.</b> Allows you to locate a file or folder for a <b>File System</b> type activity.
	<b>Link to a data source.</b> Opens the Select Link Source Dialog Box allowing you to select values for the property from a query. For details see "Select Link Source Dialog Box" on page 174.

### Action Buttons - Input/Checkpoints View



The following buttons represent actions within the **Input/Checkpoints** view:

UI Elements	Description
	<p><b>Data Drive Entire Step.</b> Data-drives all values for the step. It adds query terms to the value columns of all input properties.</p> <p>For details, see "How to Data Drive a Test Step" on page 165.</p>
	<p><b>Load from Replay.</b> Loads XML data in order to populate a service's schema with values.</p> <p>For details, see "How to Run a Test" on page 187.</p>

UI Elements	Description
	<p><b>Select Java File.</b> Opens the Call Java Class Settings Dialog Box for setting additional classpaths for the call.</p> <p>For details, see the "Call Java Class Settings Dialog Box" on page 95.</p>
	<p><b>Add New Input/Output Property.</b> Opens the Add New Input/Output Property dialog box, allowing you to define a new property and its data type.</p> <p><b>Note:</b> This only applies to the Custom Code/Property steps, REST steps, and the <b>Start/End</b> steps, or when selecting the entire canvas.</p> <p>For details, see "Coding Service Test Events" on page 261.</p>
	<p><b>Remove Property.</b> Deletes the selected property from the list.</p> <p><b>Note:</b> This only applies to the Custom Code/Property steps, REST steps, and the <b>Start/End</b> steps, or when selecting the entire canvas.</p>
<p><b>Select QTP/ST Test</b></p>	<p>For <b>Call QTP Test</b> and <b>Call Service Test</b> steps: Invokes the Open Test dialog box for selecting the test to call from this step.</p>
<p><b>Update QTP/ST Test</b></p>	<p>For <b>Call QTP Test</b> and <b>Call Service Test</b> steps: Updates the QTP or Service Test test from its original location.</p>

## SOAP Fault View

User interface elements are described below: This view is only available for Web Service or SOAP activities.

UI Elements	Description
<b>Fault is expected</b>	Indicates that a SOAP fault is expected during the test run. This setting confirms that the application did not perform a task that it was not designed to perform.
<b>XML tab</b>	<p>The SOAP envelope, containing the Header and Body of the SOAP message. In this section you enter the expected response.</p> <p>You can define <b>Any</b> type elements or values for the following properties:</p> <ul style="list-style-type: none"> <li>▶ <b>faultcode</b>. A status code indicating that the SOAP request was invalid.</li> <li>▶ <b>faultstring</b>. A response string indicating that the SOAP request was invalid.</li> <li>▶ <b>faultactor</b>. An indication of the source of the fault.</li> </ul>
<b>XPath tab</b>	<p>A list of XPATH expressions used to evaluate the SOAP response.</p> <ul style="list-style-type: none"> <li>▶  . Adds a line for an XPATH expression.</li> <li>▶  . Removes the selected entry.</li> </ul>

For more information about SOAP faults and how they can be used for Negative testing, see "Negative Testing" on page 114.

For task details, see "Configure SOAP Fault information - optional" on page 45.






## Test Variables View

User interface elements are described below: This view is only available when selecting the canvas, **Start** step, or **End** step.

UI Elements	Description
<b>User variables</b>	A list of the user variables for all profiles. When enabling the <b>Compare Profiles</b> option, the grid displays the variable values for each profile in separate columns.
<b>System variables</b>	A read-only list of common system variables and their values: ScenarioID, SystemTempDir, GroupName, TestDir, TestName, LocalHostName, OS, OSVersion, ProductDir, ProductName, ProductVer, and UserName.

### Action Buttons - Test Variables View

The following buttons represent the actions available from the **Test Variables** view (unlabeled elements are shown in angle brackets):

UI Elements	Description
	<b>Add New User Variable.</b> Adds a new user variable to the all profiles.
	<b>Remove User Variable.</b> Deletes the selected user variable from all profiles.
	<b>Add New Profile.</b> Adds a new User Variable profile to the list.
	<b>Edit Active Profile.</b> Enables you to remove or rename a profile.
	<b>Compare Profiles.</b> Displays the profiles side-by-side for comparison.
<Profile list>	A list showing the existing User Variable profiles.



## REST HTTP Input View



For REST Services, you can set the method's HTTP settings and use them as a template for that method. User interface elements are described below:

UI Elements (A-Z)	Description
<b>Input</b>	<p>The REST Input properties:</p> <ul style="list-style-type: none"> <li>▶ <b>Endpoint Address.</b> The endpoint URL for the REST service.</li> <li>▶ <b>HTTP Method.</b> GET, POST, PUT, DELETE, TRACE, OPTIONS, HEAD, and CONNECT.</li> <li>▶ <b>HTTP Version.</b> 1.0 or 1.1</li> <li>▶ <b>Request Headers.</b> Keys and their values, and arrays.</li> </ul>
<b>Checkpoints</b>	<p>The REST output properties for validation:</p> <ul style="list-style-type: none"> <li>▶ <b>HTTP Version.</b> 1.0 or 1.1</li> <li>▶ <b>Status Code.</b> An integer indicating the status code.</li> <li>▶ <b>Status Description.</b> A textual description of the status.</li> <li>▶ <b>Response Body.</b> The body returned by the server.</li> </ul> <p><b>Tip:</b> To validate a property, make sure to select its check box.</p>

## HTTP View

For REST Services and HTTP activities. User interface elements are described below (unlabeled elements are shown in angle brackets>):

UI Elements (A-Z)	Description
	<p><b>Clear.</b> Clears the contents of the:</p> <ul style="list-style-type: none"> <li>▶ Request Body (upper pane)</li> <li>▶ Checkpoint keys (lower pane)</li> </ul>
 Import File	<p>For <b>Text</b> body types, imports a file containing the body of the request.</p> <p>For <b>XML</b> body types, imports a schema file.</p>

UI Elements (A-Z)	Description
 Load XML	Loads an XML file with Body request values, that correspond to the loaded schema. ( <b>Body: XML</b> only)
	<b>Browse.</b> Allows you to load a non-XML file containing the request body ( <b>Body: File</b> only)
<checkpoint list>	A list of the checkpoints keys and their values in the bottom pane.
<request body>	The body of the request loaded through a file, pasted from a clipboard ( <b>Edit &gt; Paste</b> ), or manually entered (POST form)
<b>Body (top)</b>	The data type of the request body. <b>Text, XML, File, or Post Form.</b> <b>Default:</b> Text
<b>Body (bottom)</b>	The data type of the response body. <b>Text or XML.</b> <b>Default:</b> Text

## Events View

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

UI Elements	Description
<Events list> - default	<p>A list of events for the activity:</p> <ul style="list-style-type: none"> <li>▶ <b>CodeCheckpointEvent</b>. Triggered when the activity is executed.</li> <li>▶ <b>AfterExecuteStepEvent</b>. Triggered after the activity was executed.</li> <li>▶ <b>BeforeExecuteStepEvent</b>. Triggered before the activity is executed.</li> </ul> <p><b>Tip:</b> For details about an event, select it and choose <b>Create a default handler</b>. Refer to comments in the template.</p>
<Events list> - Web Services	<ul style="list-style-type: none"> <li>▶ AfterGenerateRequest</li> <li>▶ AfterProcessRequestSecurity</li> <li>▶ AfterProcessRequestAttachments</li> <li>▶ OnSendRequest</li> <li>▶ OnReceiveResponse</li> <li>▶ BeforeProcessResponseAttachments</li> <li>▶ BeforeProcessResponseSecurity</li> <li>▶ BeforeSaveResponse</li> <li>▶ BeforeApplyProtocolSettings</li> </ul> <p>For details about Web Services, see "Defining Services" on page 97.</p>
<b>Handler</b>	<p>The handler to call when the event is detected. Choose <b>Create a default handler</b> if no handler is available. This opens the <b>SharedUserCode.cs</b> tab for editing the handler template.</p>

For details about writing event handlers, see "Specify an event handler - optional" on page 45.



## Attachments View

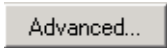

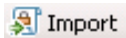
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements	Description
<Input Attachment>	<p>Input attachments for the current Web Service step:</p> <ul style="list-style-type: none"> <li>➤ <b>Type.</b> Attachment type: NONE, DIME, or MIME.</li> <li>➤ <b>Attachments.</b> A list of the input attachments and their properties: <ul style="list-style-type: none"> <li>➤ <b>Origin.</b> The file to send as an attachment.</li> <li>➤ <b>Origin Type.</b> The attachment type. <b>File</b> is default.</li> <li>➤ <b>Content Type.</b> The attachment's content type: application/zip, image/gif, image/ief, image/jpeg, image/png, image/tiff, text/plain, text/richtext, text/html, or text/xml.</li> <li>➤ <b>Content ID.</b> A unique ID for the attachment or <b>Auto</b> for an ID generated by Service Test.</li> </ul> </li> </ul> <p><b>Note:</b> To add input attachments, click <b>Attachments</b> and choose <b>Add Array Element</b> from the shortcut menu.</p>
<Output Attachment>	<p>The server response saved as an attachment.</p> <ul style="list-style-type: none"> <li>➤ <b>Attachments.</b> A list of the output attachments and their properties to validate: <ul style="list-style-type: none"> <li>➤ <b>Content.</b> The file content's checksum value. The checksum is computed by applying the MD5 hash function to the file.</li> <li>➤ <b>Content Type.</b> The attachment's content type.</li> <li>➤ <b>Content ID.</b> The unique ID of the attachment.</li> </ul> </li> </ul> <p><b>Note:</b> To add output attachments, click <b>Attachments</b> and choose <b>Add Array Element</b> from the shortcut menu.</p> <p><b>Tip:</b> To validate an output attachment, select the check boxes adjacent to the elements you want to validate. Received attachments are saved in the test's folder, with <b>bin</b> extensions.</p>

For task details, see "Add attachments to the test - optional" on page 44.

## Security Settings View

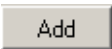
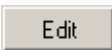
User interface elements are described below (unlabeled elements are shown in angle brackets>):

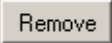
UI Elements	Description
	Opens the Advanced Settings dialog box. For details see "Advanced Settings Dialog Box" on page 137. <b>Note:</b> Not available for Web Service type scenarios.
	Saves the security scenario settings to an stss file, for use by other tests.
	Loads security scenario settings from a previously saved scenario file.
<b>Service Details</b>	A list of the available scenarios: <b>Web Service</b> , <b>WCF Service (Custom Binding)</b> , <b>WCF Service (Federation)</b> , or <b>WCF Service (Http Binding)</b> .  For information on choosing a scenario, see "Setting Security Overview" on page 98.  For user interface details about each of the scenarios, see the "Security Scenarios User Interface" on page 84.

For task details, see "How to Set Security for a Web Service" on page 118.

## Data Queries View

User interface elements are described below:


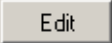
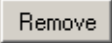
UI Elements	Description
	Opens the <b>Attach Query to Loop</b> dialog box for selecting queries from the Data Window tab.
	Opens the <b>Data Navigation</b> dialog box for setting the data policy—the way to use the data from the table. For user interface details, see "Data Navigation Dialog Box" on page 178.

UI Elements	Description
	Deletes the selected data query from the list—not from the Data Window.
<b>Data Navigation Policies</b>	<p>A list of the data queries sorted by:</p> <ul style="list-style-type: none"> <li>▶ <b>Query Name.</b> The name of the query from the Data Window.</li> <li>▶ <b>Policy.</b> A summary of the data navigation policy, for example: <b>Start at first row, forward one row, wrap around.</b></li> </ul>

For details about setting policies, see "How to Set the Navigation Properties" on page 164.

## Data Query Properties View

User interface elements are described below:

UI Elements	Description
	Opens the <b>Define New Data Relation</b> dialog box. For user interface details, see "Define/Edit New Data Relation Dialog Box" on page 181
	Opens the <b>Edit Data Relation</b> dialog box. For user interface details, see "Define/Edit New Data Relation Dialog Box" on page 181.
	Deletes the selected data relation from the list.
<b>Relations</b>	<p>A list of the data queries sorted by:</p> <ul style="list-style-type: none"> <li>▶ <b>Child Query.</b> The sheet or table to use for the query.</li> <li>▶ <b>Primary Key.</b> A column in the parent query to use as a primary key for the child relation.</li> <li>▶ <b>Foreign Key.</b> The column in the child query to use as a foreign key for the child relation.</li> </ul>

For details about defining data relations, see "Create a new child relation - optional" on page 161.

## Security Scenarios User Interface

Using the Property Sheet's Security Settings view, you can configure a security scenario for your test.

For examples, see "How to Set up Common Web Services Security Scenarios" on page 129.

This section includes:

- "Web Service Scenario Details" on page 84
- "WCF Service (Custom Binding) Scenario Details" on page 88
- "WCF Service (Federation) Scenario Details" on page 90
- "WCF Service (HTTP Binding) Scenario Details" on page 91

### Web Service Scenario Details

The Web Service scenario provides the **HTTP**, **WS-Security**, and **WS-Addressing** tabs.

#### **HTTP tab**






The **HTTP** tab lets you provide the user credentials for sending a message over HTTP. The user interface elements are described below.

UI Elements	Description
<b>User name, Password</b>	The credentials for HTTP authentication such as basic authentication, digest, or NTLM. For example, <b>User name:</b> myDomain\myUser <b>Password:</b> myPassword
<b>Client certificate</b>	The client credentials required for client certificate authentication when using two-way SSL scenarios.  The <b>Browse</b> button opens the "Select Certificate Dialog Box" on page 138.
<b>Proxy URL</b>	The URL and port of the proxy server through which the message must pass. For example, http://myProxy:8888/.

UI Elements	Description
<b>Proxy user name, Proxy password</b>	The credentials for the proxy server through which the message must pass.
<b>Keep alive</b>	Keeps the connection persistent.
<b>Connection timeout</b>	The time threshold in which to connect through the proxy server or with authentication.

## WS-Security tab

The **WS-Security** tab provides an interface to add message level security using tokens, message signatures, and encryption. The user interface elements are described below. (unlabeled elements are shown in angle brackets).

UI Elements	Description
	<b>Add a token.</b> User Name, X509, Kerberos, Kerberos2, or SAML.
	<b>Add a message signature.</b> Adds a signature to the message. This requires a token.
	<b>Add encryption.</b> Adds encryption to the message. This requires a token.
	<b>Delete.</b> Removes the security element from the list.
	<b>Up/Down.</b> Positioning tools that allow you to set the priority of the security elements. <b>Important:</b> Make sure the security elements are positioned in order of their priority.
<security element list>	A list of the tokens, message signatures, and encryptions.

UI Elements	Description
<p>&lt;Token details pane&gt; - Username tokens</p>	<p>Token details for <b>Username</b> tokens.</p> <ul style="list-style-type: none"> <li>➤ <b>Logical name.</b> An meaningful name for the token (you can use the default value).</li> <li>➤ <b>Include nonce.</b> Include a nonce in the token.</li> <li>➤ <b>User name, Password</b></li> <li>➤ <b>Password type:</b> (Text, Hash, None)</li> <li>➤ <b>Timestamp format:</b> <i>Created</i> is the default value</li> </ul>
<p>&lt;Token details pane&gt; - X509 Certificate tokens</p>	<p>Token details for <b>X509 Certificate</b> tokens:</p> <ul style="list-style-type: none"> <li>➤ <b>Logical name.</b> An meaningful name for the token.</li> <li>➤ <b>Server certificate.</b> The path of the server certificate file. The <b>Browse</b> button opens the Select Certificate Dialog Box.</li> <li>➤ <b>Reference type:</b> How the token should be referenced: <b>BinarySecurity Token</b> or <b>Reference</b>. When the certificate is used for encryption, for example, a service certificate, use <b>Reference</b>. When using it for a signature (for example, a certificate with your private key) choose <b>BinarySecurity Token</b>.</li> </ul>
<p>&lt;Token details pane&gt; - Kerberos tokens</p>	<p>Token details for <b>Kerberos</b> tokens:</p> <ul style="list-style-type: none"> <li>➤ <b>Logical Name.</b> An meaningful name for the token.</li> <li>➤ <b>Host.</b> The host name of the server to which you want to authenticate. In most cases, it is the host portion of the service URL.</li> <li>➤ <b>Domain.</b> The Windows domain of the server to which you want to authenticate.</li> </ul>
<p>&lt;Token details pane&gt; -SAML tokens</p>	<p>Token details for <b>SAML</b> tokens:</p> <ul style="list-style-type: none"> <li>➤ <b>Load from file.</b> Allows you to browse for a SAML certificate.</li> <li>➤ <b>Client certificate.</b> The path of the client certificate file. The <b>Browse</b> button opens the Select Certificate Dialog Box.</li> <li>➤ <b>Certificate reference type.</b> How the certificate should be referenced— by X509 Data or RSA.</li> </ul>

UI Elements	Description
<Signature details pane>	<p>The details of the digital signature used to secure the token.</p> <ul style="list-style-type: none"> <li>▶ <b>Use token.</b> The token to use for signing, usually an X.509 type. Select from the dropdown list of all added tokens.</li> <li>▶ <b>Target token.</b> The token you want to sign. Select from the dropdown list of all added tokens (optional). With most services, this field should be left empty.</li> <li>▶ <b>Target path.</b> An XPath which specifies which parts in the message to sign. If left blank, the elements selected in the <b>Signature options</b> field are signed.</li> <li>▶ <b>Canonicalization algorithm.</b> A URL for the algorithm to use for canonicalization. A dropdown provides common algorithms. If you are unsure which value to use, keep the default.</li> <li>▶ <b>Transform algorithm.</b> A URL for the Transform algorithm to apply to the message signature. A dropdown provides common algorithms. If you are unsure which value to use, keep the default.</li> <li>▶ <b>Inclusive namespaces list.</b> A list of prefixes to be treated as inclusive (optional).</li> <li>▶ <b>Signature options.</b> Which SOAP elements should be signed: Body, Timestamp, and WS-Addressing.</li> </ul> <p><b>Tip:</b> You can also specify an XPATH in the message which you want to be signed. e.g. <code>//*[local-name()='Body']</code></p>

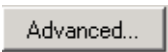
UI Elements	Description
<Encryption details pane>	<p>The details of the encryption token.</p> <ul style="list-style-type: none"> <li>▶ <b>Use token.</b> The token to use for encryption, usually an X.509 type. A dropdown box provides a list of all available tokens.</li> <li>▶ <b>Target token.</b> The name of the encrypted token. A dropdown box provides a list of all added tokens. With most services, this field should be left empty.</li> <li>▶ <b>Target path.</b> An XPath which indicates the parts of the message to encrypt. If left blank, only the SOAP body is encrypted.</li> <li>▶ <b>Type.</b> Whether to encrypt the whole destination <b>Element</b> or only its <b>Content</b>.</li> <li>▶ <b>Key algorithm.</b> The algorithm to use for the encryption of the session key: <b>RSA15</b> or <b>RSOAEP</b>.</li> <li>▶ <b>Session algorithm.</b> The algorithm to use for the encryption of the SOAP message. A dropdown list provides common values.</li> </ul>

### WS-Addressing tab

The **WS-Addressing** tab indicates whether WS-Addressing is used by the scenario, and if so, its version number.

### WCF Service (Custom Binding) Scenario Details

Use this scenario to test WCF services which require security or transport configurations. User interface elements are described below (unlabeled elements are shown in angle brackets). For details see "WCF Service (Custom Binding) Scenarios" on page 103.

UI Elements	Description
	Opens the Advanced Settings dialog box. For details see "Advanced Settings Dialog Box" on page 137.
<b>Transport</b>	<b>Transport type.</b> HTTP, HTTPS, TCP, NamedPipe, or AutoSecuredHTTP.



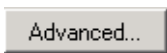
UI Elements	Description
<b>Encoding</b>	<b>Message encoding.</b> Text, MTOM, or WCF Binary
<b>Security</b>	<p><b>Authentication mode.</b> For a list of modes, see "WCF Service (Custom Binding) Scenarios" on page 103.</p> <p><b>Note:</b> If you choose a the SecureConversation mode, select a <b>Bootstrap Policy</b>.</p>
<b>Net Security</b>	The type of stream security: <b>None</b> , <b>Windows stream</b> or <b>SSL stream</b> .
<b>Reliable Messaging</b>	Enables <b>Reliable Messaging</b> in <b>Ordered Or Non-ordered</b> format.
<b>Identities</b>	<p>The identity information for the bindings and certificate:</p> <ul style="list-style-type: none"> <li>➤ <b>Username and Password</b></li> <li>➤ <b>Server /Client Certificates</b></li> <li>➤ <b>Expected DNS, SPN, and UPN</b></li> </ul>
<b>Client Windows Identity</b>	<p>The identity information for the client windows:</p> <ul style="list-style-type: none"> <li>➤ <b>Current User</b> logged onto the machine</li> <li>➤ <b>Custom User</b> with the following credentials: <ul style="list-style-type: none"> <li>➤ <b>Username</b></li> <li>➤ <b>Password</b></li> <li>➤ <b>Domain</b></li> </ul> </li> </ul>

For details about additional security settings, see the "Advanced Settings Dialog Box" on page 137.

## WCF Service (Federation) Scenario Details

In the **WCF Service (Federation)** scenario, the client authenticates against the STS (Security Token Service) to obtain a token. The client uses the token to authenticate against the application server.

User interface elements are described below (unlabeled elements are shown in angle brackets>). For details see "WCF Service (Federation) Scenarios" on page 105.

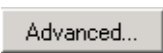
UI Elements	Description
	Opens the Advanced Settings dialog box. For details see "Advanced Settings Dialog Box" on page 137.
<b>Server Transport</b>	<b>Transport.</b> The transport type: <b>HTTP</b> or <b>HTTPS</b> . <b>Encoding.</b> The server's encoding policy: <b>Text</b> or <b>MTOM</b> .
<b>Security</b>	<b>Authentication mode.</b> The authentication mode for the scenario. For a list of modes, see "WCF Service (Federation) Scenarios" on page 105. <b>Note:</b> If you choose a the SecureConversation mode, select a <b>Bootstrap Policy</b> .
<b>Identities</b>	The identity information for the bindings and certificate: <ul style="list-style-type: none"> <li>▶ <b>Server Certificate.</b> The <b>Browse</b> button opens the Select Certificate Dialog Box.</li> <li>▶ <b>Expected DNS.</b> The expected identity of the server in terms of its DNS. This can be <b>localhost</b>, an IP address, or server name.</li> </ul>
<b>STS (Security Token Service)</b>	Information about the security token: <ul style="list-style-type: none"> <li>▶ <b>Issuer address.</b> The address of the issuer of the STS. This can be <b>localhost</b>, an IP address, or a server name.</li> <li>▶ <b>Referenced binding.</b> The scenario which references the binding that contacts the STS (Security Token Service).</li> </ul>

For details about additional security settings, see the "Advanced Settings Dialog Box" on page 137.

## WCF Service (HTTP Binding) Scenario Details

In the **WCF Service (Http Binding)** scenario, you can choose from several types of authentication: None, Windows, Certificate, or Username (message protection). For details see "WCF Services (Http Binding) Scenarios" on page 106.

User interface elements are described below (unlabeled elements are shown in angle brackets>) by the client authentication types:

UI Elements	Description
	Opens the Advanced Settings dialog box. For details see "Advanced Settings Dialog Box" on page 137.
<b>Client authentication type</b>	Authentication type: <b>None</b> , <b>Windows</b> , <b>Certificate</b> , or <b>Username</b> .

### Client Authentication Type — None

UI Elements	Description
<b>Negotiate server credentials</b>	Negotiates the Web Service's certificate with the server. You can also provide the server's DNS information.
<b>Specify service certificate</b>	The location of the service's certificate. If you select this option, the <b>Negotiate service credentials</b> option is not relevant.  For more information, see the "Select Certificate Dialog Box" on page 138.
<b>Expected server DNS</b>	The expected identity of the server in terms of its DNS. This can be <b>localhost</b> , an IP address, or a server name. It can also be the common name by which the certificate was issued.

## Client Authentication Type — Windows

UI Elements	Description
<b>Expected server identity</b>	The expected server identity method SPN or UPN.
<b>Client Windows identity</b>	<p>The identity information for the client windows:</p> <ul style="list-style-type: none"> <li>➤ <b>Current User</b> logged into the machine</li> <li>➤ <b>Custom User</b> with the following credentials: <ul style="list-style-type: none"> <li>➤ <b>Username</b></li> <li>➤ <b>Password</b></li> <li>➤ <b>Domain</b></li> </ul> </li> </ul>

## Client Authentication Type — Certificate

UI Elements	Description
<b>Client certificate</b>	The location of the client certificate. The <b>Browse</b> button opens the Select Certificate Dialog Box.
<b>Negotiate server credentials</b>	Negotiates the Web Service's certificate with the server. You can also provide the server's DNS information.
<b>Specify service certificate</b>	<p>The location of the service's certificate. If you select this option, the <b>Negotiate service credentials</b> option is not relevant.</p> <p>For more information, see the "Select Certificate Dialog Box" on page 138.</p>
<b>Expected server DNS</b>	The expected identity of the server in terms of its DNS. This can be <b>localhost</b> , an IP address, or a server name. It can also be the common name by which the certificate was issued.

## Client Authentication Type — Username (message protection)



UI Elements	Description
<b>Username, Password</b>	The authentication credentials of the client.
<b>Negotiate server credentials</b>	Negotiates the Web Service's certificate with the server. You can also provide the server's DNS information.
<b>Specify service certificate</b>	The location of the service's certificate. If you select this option, the <b>Negotiate service credentials</b> option is not relevant.  For more information, see the "Select Certificate Dialog Box" on page 138.
<b>Expected server DNS</b>	The expected identity of the server in terms of its DNS. This can be <b>localhost</b> , an IP address, or a server name. It can also be the common name by which the certificate was issued.

### Output Tab

Enables you to view the output messages that were generated while compiling and running the test.

<b>To Access</b>	<b>View &gt; Output</b> (Ctrl+Alt+O)
<b>Important Information</b>	Click on a row to navigate to the code that generated the output message.
<b>Relevant Tasks</b>	"How to Run a Test" on page 187

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

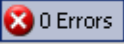





GUI Element (A-Z)	Description
	<b>Clear All.</b> Clears all the information in the message window.
	<b>Toggle Word Wrap.</b> Wraps the text of each message onto the next line
<Show Output from>	Dropdown list for selecting the output to display: <b>Build, Design, Debug, or Run.</b> <b>Default:</b> Run
<message window>	Displays all build and debug information issued to the log during the test run.

## Errors Tab

Enables you to view a list of the errors generated during the test run, and their level of severity: message, warning, or error.


<b>To Access</b>	Click <b>Errors</b> tab in bottom pane or select <b>View &gt; Errors</b> or Ctrl+Alt+K
<b>Important Information</b>	<ul style="list-style-type: none"> <li>▶ Click on a row to jump to the line in the code that generated the error.</li> <li>▶ Click any column header, for example <b>Line</b> or <b>File</b>, to sort by that criteria.</li> </ul>
<b>Relevant Tasks</b>	"How to Run a Test" on page 187
<b>See Also</b>	"Alerts" on page 26

The following elements are included:

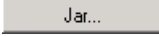
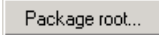
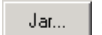
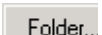
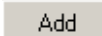

UI Elements (A-Z)	Description
	Displays the errors detected during the test run.
	Displays the warnings detected during the test run.
	Displays the informational messages detected during the test run.
<b>! (Exclamation Point)</b>	Message Type: <ul style="list-style-type: none"> <li>▶  Error</li> <li>▶  Warning</li> <li>▶  Informative message</li> </ul>
<b>Description</b>	Description of the error, warning or message and advice on how to fix the problem.
<b>File</b>	The file that generated the error, warning, or message.
<b>Line</b>	The line number in which the error was detected.
<b>Path</b>	The complete path of the file that generated the error.

## Call Java Class Settings Dialog Box

This dialog box enables you to configure the settings for a call to a Java class.

<b>To access</b>	<ul style="list-style-type: none"> <li>▶ Add a <b>Call Java Class</b> step to the canvas.</li> <li>▶ In the Property Sheet, open the Java Call Input Output Input/Checkpoints view  .</li> <li>▶ Click the <b>Select Java File</b> button.</li> </ul>
<b>Relevant tasks</b>	"How to Create a Call to a Java Class" on page 49

User interface elements are described below:

UI Elements	Description
<b>Select location</b>	<p>The location of the Java class.</p> <ul style="list-style-type: none"> <li>▶  - Opens the <b>Select Jar File</b> dialog box.</li> <li>▶  - Opens the <b>Browse for Folder</b> dialog box to select a root folder.</li> </ul>
<b>Select Java class</b>	<p>The Java class to associate with the <b>Call Java Class</b> step. When you click Browse, Service Test shows the classes in the specified jar file.</p>
<b>System classpath</b>	<p>The paths you defined in the Test Settings for system classpaths.</p> <p>To modify this setting, click in the canvas (outside the Test Flow or Loop) and open the Property Sheet's <b>General</b> view. Modify the Test Setting's <b>JVM &gt; Classpath</b> node.</p>
<b>Additional classpaths box</b>	<p>Additional classpaths to associate with the <b>Call Java Class</b> step.</p> <ul style="list-style-type: none"> <li>▶  - Opens the <b>Select Jar File</b> dialog box.</li> <li>▶  - Opens the <b>Browse for Folder</b> dialog box to select a folder containing the .class file.</li> </ul>
<b>&lt;additional classpath list&gt;</b>	<p>A list of jar files or folders containing the additional classes.</p> <ul style="list-style-type: none"> <li>▶  - Adds the contents of the <b>Additional Classpath</b> box to the additional classpath list.</li> <li>▶  - Removes the selected entry from the list.</li> </ul>



# 3

---

## Defining Services

This chapter includes:

### Concepts

- ▶ [Preparing Service References](#) on page 98
- ▶ [Setting Security Overview](#) on page 98
- ▶ [Security Scenarios Overview](#) on page 99
- ▶ [WCF Scenario Settings](#) on page 103
- ▶ [Advanced Security Settings](#) on page 109
- ▶ [Negative Testing](#) on page 114

### Tasks

- ▶ [How to Import a WSDL-Based Web Service](#) on page 116
- ▶ [How to Set Security for a Web Service](#) on page 118
- ▶ [How to Create a Simple Test for a REST Service](#) on page 122
- ▶ [How to Create a Template-Based REST Service](#) on page 125
- ▶ [How to Set up Common Web Services Security Scenarios](#) on page 129

### Reference

- ▶ [Import Service User Interface](#) on page 133
- ▶ [Security User Interface](#) on page 136

[Tips and Guidelines](#) on page 139

---

---

## Concepts

---

---

### Preparing Service References

Service Test lets you import or define services for the purpose of creating high-level tests.

For Web Services, you typically begin creating a test by importing a WSDL file. This file provides a structure for the test, since it describes the service in terms of its elements, argument values and properties. For information on importing a WSDL, see "How to Import a WSDL-Based Web Service" on page 116.

If your service document is unique and cannot be imported in the normal way, you can use the SOAP Request activity to send a manual SOAP request to the server. action - when ST doesn't support the service/wSDL type.

For REST services, you define the metadata manually by modelling the service and defining your own methods. For task details, see "How to Create a Template-Based REST Service" on page 125.

### Setting Security Overview

When building Web Service applications, there is a challenge in building scalable applications that are secure. You can secure Web Services by having the message sent over a secure transport, such as Secure Sockets Layer (SSL), or by applying security at the message level, also known as **WS-Security**.

For testing a secured service, answering the following questions will help you define your security scenario.

- ▶ Is there transport security, such as SSL? What is the HTTPS URL?
- ▶ Is basic authentication required?
- ▶ Is mutual authentication required?
- ▶ What type of security is required in the SOAP header?

For task details, see "How to Set Security for a Web Service" on page 118.

## Security Scenarios Overview

Service Test provides several built-in scenarios for configuring security in Web Service calls.

A security scenario represents a typical environment of a Web Service security and contains information such as security, encoding, proxy, certificates, and so forth.

A default **Web Service** scenario can be used for most Web services. It allows you to configure both transport and message-level security. Service Test support for message-level security lets you manually configure the security elements such as tokens, message signatures, and encryption. For details, see "Web Service Security Scenario" on page 100.

WCF scenarios allows you to configure security for HTTP or custom bindings and work with advanced specifications, such as **WS-SecureConversation**.

---

**Note:** When testing Web Services that require message-level security, the Web Service security scenario only supports SOAP version 1.1. For SOAP 1.2 use a WCF type scenario.

---

The built-in security scenarios are

- Web Service Security Scenario
- WCF Service (Custom Binding) Scenarios
- WCF Service (Federation) Scenarios
- WCF Services (Http Binding) Scenarios

**Use the default Web Service scenario for:**

- Simple Web Services where no advanced standards are required.

- ▶ Web services using HTTP transport level security or message level security (WS-Security).

**Use a WCF scenario to:**

Test WCF Services that utilize advanced security and WS-Specifications. Such services can be written in various platforms such as WCF (Windows Communication Foundation), Metro (WSIT), and Axis2. Service Test also supports proprietary standards and transports.

 **Web Service Security Scenario**

The default Web Service scenario is based on the WS-Security specification. This scenario lets you place security credentials in the actual SOAP message.

When a SOAP message sender sends a request, the security credentials, known as **tokens**, are placed in the SOAP message. When the Web server receives the SOAP request, it does not need to send additional requests to verify the integrity of the sender. The server verifies that the credentials are authentic before letting the Web Service execute the application. By not having to go back to the source of the credentials, the application's scalability improves significantly.

To further secure Web Services, it is common to use digital signatures or encryption for the SOAP messages. Digitally signing a SOAP message verifies that the message has not been altered during transmission. Encrypting a SOAP message helps secure a Web Service by making it difficult for anyone other than the intended recipient to read the contents of the message.

The default Web Service security scenario contains the following tabs: **HTTP**, **WS-Security**, and **WS-Addressing**. The **HTTP** tab lets you configure the transport security, the **WS-Security** tab handles the message-level security, and the **WS-Addressing** tab sets the addressing version.

For user interface details, see "Security Settings View" on page 35.

**Transport Level Security**

The transport level security includes the authentication and proxy server information. You can also specify Keep Alive preferences and connection timeout. For user interface details, see "Security Settings View" on page 35.

## Message Level Security

The **WS-Security** tab lets you set the message level security through tokens, signatures and encryption.

To support WS-Security, Service Test allows you to create security tokens for your script. You can create multiple tokens and set their properties. After creating a token, you use it to sign or encrypt a SOAP message.

The Web Services security mechanism associates security tokens with messages. This mechanism supports several security token formats to accommodate a variety of authentication requirements. For example, a client might need to provide a proof of identity or a security certificate.

The available tokens are **Username Token**, **X.509 Certificate Token**, **Kerberos Token**, **Kerberos2 Token**, and **SAML**.

- ▶ **User Name Token.** The **User Name and Password** token contains user identification information for the purpose of authentication: **User Name** and **Password**. You can also specify Password Options, indicating how to send the password to the server for authentication: **Text**, **None**, or **Hash**.
- ▶ **X.509 Certificate Token.** This security token is a token based on an X.509 certificate. To obtain a certificate, you can either purchase it from a certificate authority, such as VeriSign, Inc. or set up your own certificate service to issue a certificate. Most Windows servers support the public key infrastructure (PKI) which enable you to create certificates. You can then have it signed by a certificate authority or use an unsigned certificate.
- ▶ **Kerberos Token/Kerberos2 Ticket.** (for Windows 2003 or XP SP1 and later) The Kerberos protocol is used to mutually authenticate users and services on an open and unsecured network. Using shared secret keys, it encrypts and signs user credentials. A third party, known as a KDC (Kerberos Key Distribution Center), authenticates the credentials. After authentication, the user may request a service ticket to access one or more services on the network. The ticket includes the encrypted, authenticated identity of the user. The tickets are obtained using the current user's credentials.

The primary difference between the Kerberos and Kerberos2 tokens is that Kerberos2 uses the Security Support Provider Interface (SSPI), so it does not require elevated privileges to impersonate the client's identity. In addition, the Kerberos2 security token can be used to secure SOAP messages sent to a Web Service running in a Web farm.

- ▶ **SAML Token.** SAML is an XML standard for exchanging security-related information, called assertions, between business partners over the Internet. The assertions can include attribute statements, authentication, decision statements, and authorization decision statements.

SAML uses brokered authentication with a security token issued by STS (Security Token Service). The STS is trusted by the client and the Web Service to provide interoperable security tokens. SAML tokens are important for Web Service security because they provide cross-platform interoperability and a means of exchanging information between clients and services that do not reside within a single security domain.

### **Message Signatures and Encrypted Data**

When you add a security token to a SOAP message, it is added to the SOAP message in the form of an XML element in the WS-Security SOAP header.

The message, however, is exposed and therefore requires additional security. This is especially true when the credentials, including the password, are sent in plain text as it is with role-based security.

The two methods used to secure the data are message signatures and message encryption.

- ▶ **Message Signatures.** Message Signatures are used by the recipients to verify that messages were not altered since their signing. The signature is in the form of XML within the SOAP message. The recipient checks the signature to make sure it is valid.
- ▶ **Message Encryption.** Although the XML message signature offers a mechanism for verifying that the message has not been altered since it was signed, it does not encrypt the SOAP message—the message is still plain text in XML format. To secure the message in order that it should not be exposed, you encrypt it, making it difficult for an intruder to view and obtain a user's password.

For task details, see "How to Set Security for a Web Service" on page 118.

## WCF Scenario Settings

This section describes the values required for the WCF security scenarios: It includes:

- "WCF Service (Custom Binding) Scenarios" on page 103
- "WCF Service (Federation) Scenarios" on page 105
- "WCF Services (Http Binding) Scenarios" on page 106

### WCF Service (Custom Binding) Scenarios

The **WCF Service (Custom Binding)** scenario enables the highest degree of customization. Since it is based upon WCF **customBinding**, it allows you to test most WCF services, along with services on other platforms such as Java that use WS - *<spec\_name>* specifications.

Use the **WCF Service (Custom Binding)** scenario to configure a scenario that does not comply with any of the predefined security scenarios. You can use Service Test's default settings or customize the following elements:

- **Transport.** HTTP, HTTPS, TCP, or NamedPipe
- **Encoding.** Text, MTOM, or WCF Binary

Specify the following security information:

- ▶ **Authentication mode.** None, AnonymousForCertificate, AnonymousForSslNegotiated, CertificateOverTransport, Kerberos, KerberosOverTransport, MutualCertificate, MutualSslNegotiated, SecureConversation, SspiNegotiated, UserNameForCertificate, UserNameForSslNegotiated, UserNameOverTransport, or SspiNegotiatedOverTransport.
- ▶ **Bootstrap policy.** For the **SecureConversation** authentication mode, you can specify a bootstrap policy: AnonymousForCertificate, AnonymousForSslNegotiated, CertificateOverTransport, Kerberos, KerberosOverTransport, MutualCertificate, MutualSslNegotiated, SspiNegotiated, UserNameForCertificate, UserNameForSslNegotiated, UserNameOverTransport, or SspiNegotiatedOverTransport.
- ▶ **Net security.** For **TCP** and **Named Pipe** transports, you can set the network security. Select **None**, **Windows stream security**, or **SSL stream security**. For services with HTTP transport, leave the default value, **None**. To enable SSL for HTTP, choose the HTTPS transport.

### Reliable Messaging

If your Web Service uses **Reliable messaging**, enable the option, and select **Ordered** or **Not Ordered**.

### Identities

Your security settings may require you to provide identity details for either the client and server, or both of them.

An example of identity details for the client, are user name/password or an **X.509** certificate.

For identity information, provide one or more authentication details as required by the service:



**Username, Password, Server certificate, Client certificate**, or a custom Windows identity. For details about choosing a certificate, see "Select Certificate Dialog Box" on page 138.

Some scenarios require you to declare the expected identity of the server in terms of its DNS, SPN, or UPN identity.

- **DNS.** Provide the name of a server or use localhost.
- **SPN.** Provide the SPN identity in the domain\machine format.
- **UPN.** Provide the UPN identity in the user@domain format.

After setting the basic values, you can set advanced attributes as described in "Advanced Security Settings" on page 109.

---

**Note:** For WSE3 security configurations, use the **WCF Service (Custom Binding)** Scenario. For details, see "How do I test a WCF service?" on page 139.

---

## **WCF Service (Federation) Scenarios**

In the **WCF Service (Federation)** scenario, the client authenticates against the STS (Security Token Service) to obtain a token. The client uses the token to authenticate against the application server.

Therefore, two bindings are needed, one against the STS and another against the application server.

Click the Browse button in the **Referenced binding** field to define an STS binding. Then, fill in the following Security Details for the application server's security scenario:

### **Server**

- **Transport.** HTTP or HTTPS
- **Encoding.** Text or MTOM

## Security

- **Authentication mode.** IssuedToken, IssuedTokenForCertificate, IssuedTokenForSslNegotiated, IssuedTokenOverTransport, or SecureConversation.
- **Bootstrap policy.** IssuedToken, IssuedTokenForCertificate, IssuedTokenForSslNegotiated, or IssuedTokenOverTransport.

## Identity

- **Server certificate.** Browse for a server certificate. For more information, see the "Select Certificate Dialog Box" on page 138.
- **Expected DNS.** The expected identity of the server in terms of its DNS. This can be **localhost**, an IP address, or a server name.

## STS (Security Token Service)

- **Issuer address.** The address of the issuer of the STS. This can be **localhost**, an IP address, or a server name.
- **Referenced binding.** The scenario which references the binding that contacts the STS (Security Token Service).

For task details, see "Set up a WCF Federation scenario - optional" on page 120.

## WCF Services (Http Binding) Scenarios

---

**Note:** The **WCF Service (Http Binding)** scenario only supports the testing of WCF services which utilize *wsHttpBinding* and incorporate some level of security. To test WCF services that use *wsHttpBinding* but have no security, use the **WCF Service (Custom Binding)** scenario.

---

In the **WCF Service (Http Binding)** scenario, you can choose from several types of authentication: None, Windows, Certificate, or Username (message protection).

For all of the authentication types, you can apply advanced settings as described in "Advanced Security Settings" on page 109.

### No Authentication (Anonymous)

In this scenario, the client uses the server's certificate to encrypt a message; there is no client authentication.

You specify only one of the following settings:

- ▶ **Negotiate service credentials.** Negotiate the Web Service's certificate with the server.
- ▶ **Specify service certificate.** Browse for a service certificate. For more information, see "Select Certificate Dialog Box" on page 138. If you select this option, the **Negotiate service credentials** option is not available.

You can also provide the server's DNS information.

- ▶ **Expected server DNS.** The expected identity of the server in terms of its DNS. This can be **localhost**, an IP address, or a server name. It can also be the common name by which the certificate was issued.

### Windows Authentication

This WCF scenario uses Windows Authentication. If you are testing a WCF service that has not been customized and uses the default configuration, use this type of scenario.

You declare the expected identity of the *server* in terms of its **SPN** or **UPN** identities.

You declare the expected identity of the *client* in terms of the current user credentials or those of a custom user. If you choose the identity of a custom user, you provide, a Username, Password, and Domain.

## Certificate Authentication

In this WCF WSHttpBinding scenario, the client uses the server's X.509 certificate to encrypt the message and its own certificate for a signature.

Specify only one of the following settings:

- ▶ **Negotiate service credentials.** Negotiate the Web Service's certificate with the server.
- ▶ **Specify service certificate.** Browse for a service certificate. For details, see "Select Certificate Dialog Box" on page 138. If you select this option, the **Negotiate service credentials** option is not available.

Provide the DNS information:

- ▶ **Expected server DNS.** The expected identity of the server in terms of its DNS. This can be **localhost**, an IP address, or a server name. It can also be the common name by which the certificate was issued.

## Username Authentication (Message Protection)

In this WCF WSHttpBinding scenario, the client uses the server's X.509 certificate to encrypt the message, and sends a user name and password to authenticate itself.

Specify the following settings:

- ▶ **Username. Password.** The client's user name and password credentials.

Specify only one of the following settings:

- ▶ **Negotiate service credentials.** Negotiate the Web Service's certificate with the server.
- ▶ **Specify service certificate.** Browse for a service certificate. For details, see "Select Certificate Dialog Box" on page 138. If you select this option, the **Negotiate service credentials** option is not available.

You can also provide the server's DNS information.

- ▶ **Expected server DNS.** The expected identity of the server in terms of its DNS. This can be **localhost**, an IP address, or a server name. It can also be the common name by which the certificate was issued.

## Advanced Security Settings

This section describes the Advanced scenario settings to customize a **WCF Service (Custom Binding)** scenario in the areas of **Encoding, Advanced Standards, Security, or HTTP and Proxy**.

Not all settings are relevant for all scenarios, so some of them might be disabled or hidden depending on the scenario.

This section includes:

- "Encoding" on page 109
- "Advanced Standards" on page 109
- "Security" on page 110
- "HTTP and Proxy" on page 113

### Encoding

The Encoding tab lets you indicate the type of encoding to use for the messages: **Text, MTOM, or WCF Binary**. The default is **Text** encoding.

For each of these encoding methods, you can choose a version of WS-Addressing:

- None
- WSA 1.0
- WSA 04/08

### Advanced Standards

This tab lets you configure advanced WS- standards, such as Reliable Messaging and the Via address option.

If your service implements the **WS-ReliableMessaging** specification, you can enable the **Reliable Messaging** option and set the following options:

- **Reliable messaging ordered.** indicates whether the reliable session should be ordered

- ▶ **Reliable messaging version.** WSReliableMessagingFebruary2005 or WSReliableMessaging11

### Via Address

In certain instances, you may need to send a message to an intermediate service that submits it to the actual server. This may also apply when you send the message to a debugging proxy. This corresponds to the WCF **clientVia** behavior.

In such cases it may be useful to separate the physical address to which the message is actually sent, from the logical address for which the message is intended. The logical address may be the physical address of the final server or any name. It appears in the SOAP message as follows:

```
<wsa:Action>http://myLogicalAddress<wsa:Action>
```

The logical address is retrieved from the user interface. By default, it is the address specified in the WSDL. You can override this address from the Manage Services dialog box.

### Security

The Advanced security settings correspond to the **WS-Security** specifications.

For security scenarios that are based upon WCF WSHttpBinding, you can indicate the following settings:

- ▶ **Enable secure session.** Establish a security context using the WS-SecureConversation standard.
- ▶ **Negotiate service credentials.** Allow WCF proprietary negotiations to negotiate the service's security.

For Custom Binding WCF type scenarios, you can set the default algorithm suite and protection level:

Attribute	Meaning	Possible Values
<b>Default algorithm suite</b>	The algorithm to use for symmetric/asymmetric encryption.  These are the values from the SecurityAlgorithmSuite configuration in WCF:	<ul style="list-style-type: none"> <li>➤ Basic128</li> <li>➤ Basic128Rsa15</li> <li>➤ Basic128Sha256</li> <li>➤ Basic128Sha256Rsa15</li> <li>➤ Basic192</li> <li>➤ Basic192Rsa15</li> <li>➤ Basic192Sha256</li> <li>➤ Basic192Sha256Rsa15</li> <li>➤ Basic256</li> <li>➤ Basic256Rsa15</li> <li>➤ Basic256Sha256</li> <li>➤ Basic256Sha256Rsa15</li> <li>➤ TripleDes</li> <li>➤ TripleDesRsa15</li> <li>➤ TripleDesSha256</li> <li>➤ TripleDesSha256Rsa15</li> </ul>
<b>Protection level</b>	Should the SOAP Body be encrypted/signed	None, Sign, and Encrypt And Sign (default)

You can customize the security settings in greater detail. The following table describes the options and their values:

Attribute	Meaning	Possible Values
<b>Message protection order</b>	The order for signing and encrypting	<ul style="list-style-type: none"> <li>➤ Sign Before Encrypt</li> <li>➤ Sign Before Encrypt-And Encrypt Signature</li> <li>➤ Encrypt Before Sign</li> </ul>
<b>Message security version</b>	The WS-Security security version	A list of the current versions
<b>Security header layout</b>	The layout for the message header	<ul style="list-style-type: none"> <li>➤ Strict</li> <li>➤ Lax</li> <li>➤ LaxTimeStampFirst</li> <li>➤ LaxTimeStampLast</li> </ul>
<b>Key entropy mode</b>	The entropy mode for the security key.	<ul style="list-style-type: none"> <li>➤ Client Entropy</li> <li>➤ Security Entropy</li> <li>➤ Combined Entropy</li> </ul>

You can enable or disable the following options:

- ▶ **Require derived keys.** Indicates whether or not to require derived keys.
- ▶ **Require security context cancellation.** Disabling this option implies that stateful security tokens will be used in the **WS-SecureConversation** session (if enabled).
- ▶ **Include timestamp.** Includes a timestamp in the header.
- ▶ **Allow serialized signing token on reply.** Enables the reply to send a serialized signing token.
- ▶ **Require signature confirmation.** Instructs the server to send a signature confirmation in the response.

For X.509 certificates, you can specify values for the following items:

Attribute	Meaning	Possible Values
<b>X509 Inclusion Mode</b>	When to include the X509 certificate	<ul style="list-style-type: none"> <li>▶ Always to Recipient</li> <li>▶ Never</li> <li>▶ Once</li> <li>▶ Always To Initiator</li> </ul>
<b>X509 Reference Style</b>	How to reference the certificate	<ul style="list-style-type: none"> <li>▶ Internal</li> <li>▶ External</li> </ul>
<b>X509 require derived keys</b>	Should X509 certificates require derived keys	<ul style="list-style-type: none"> <li>▶ Enable - Yes</li> <li>▶ Disable - No</li> </ul>
<b>X509 key identifier clause type</b>	The type of clause used to identify the X509 key.	<ul style="list-style-type: none"> <li>▶ Any</li> <li>▶ Thumbprint</li> <li>▶ Issuer Serial</li> <li>▶ Subject Key Identifier</li> <li>▶ Raw Data Key Identifier</li> </ul>



## HTTP and Proxy

This tab lets you set the HTTP and Proxy information for your test.

### HTTP (S) Transport

The following table describes the HTTP(S) Transport options:

Option	Meaning	Possible Values
<b>Transfer mode</b>	The transfer method for requests/responses	Buffered, Streamed, Streamed Request, Streamed Response
<b>Max response size (KB)</b>	The maximum size of the response before being concatenated	Default 65 KB
<b>Allow cookies</b>	Enable cookies	Enabled / Disabled
<b>Keep-Alive Enabled</b>	Enable keep-alive connections	Enabled / Disabled
<b>Authentication scheme</b>	HTTP authentication method	None, Digest, Negotiate, NTLM, Integrated Windows Authentication, Basic, Anonymous
<b>Realm</b>	The realm of the authentication scheme	Any URL
<b>Require client certificate</b>	For SSL transport, require a certificate	Enabled / Disabled

## Proxy Information

If the Web service's transport uses a proxy server, you can specify its details in the **Security** tab. The following table describes the proxy options:

Option	Meaning	Possible Values
<b>Use default web proxy</b>	Use machine's default proxy settings	Enabled / Disabled
<b>Bypass proxy on local</b>	Ignore proxy when the service is on the local machine	Enabled / Disabled
<b>Proxy address</b>	the proxy server	Any URL
<b>Proxy authentication scheme</b>	HTTP authentication method on Proxy	Digest, Negotiate, NTLM, Basic, Anonymous

## Negative Testing

When performing a functional test for your Web Service, you should approach the testing in a variety of ways. The most common type of testing is called **Positive Testing**—checking that the service does what it was designed to do.

In addition, you should perform **Negative Testing**, to confirm that the application did not perform a task that it was not designed to perform. In those cases, you need to verify that the application issued an appropriate error—a SOAP Fault.

To illustrate this, consider a form accepting input data—you apply positive testing to check that your Web Service has properly accepted the name and other input data. You apply negative testing to make sure that the application detects an invalid character, for example a letter character in a telephone number.

When your service send requests to the server, the server responds in one of the following ways:

- ▶ **SOAP Result.** A SOAP response to the request
- ▶ **SOAP Fault.** A response indicating that the SOAP request was invalid. Negative Testing applies only to SOAP faults.
- ▶ **HTTP Error.** An HTTP error, such as Page Not Found, unrelated to Web Services.

Service Test can check for a standard SOAP result or a SOAP fault response. For example, if your Web Service attempts to access a Web page that cannot be found, it will issue a 404 HTTP error. Using negative testing you indicate that you expect a SOAP fault. In this case, the test run will fail if the service accesses a *valid* Web page.

Service Test's Property Sheet lets you provide values for the SOAP fault header and body. You can enter **faultcode**, **faultstring**, and **faultactor** values as well as custom properties using **Any** type parameters. Using the Checkpoint mechanism, you can validate these values and view the results in the Run Results Viewer.

For details on setting up the SOAP fault values, see "SOAP Fault View" on page 76.

---

---

## Tasks

---

---

### **How to Import a WSDL-Based Web Service**

This task describes how to import a Web Service into your project.


This task includes the following steps:

- "Prerequisites" on page 116
- "Open the Import/ Select Service dialog box" on page 116
- "Select a source" on page 117
- "Set connection settings for URL/UDDI imports - optional" on page 118
- "Complete the Import" on page 118

#### **1 Prerequisites**

If you intend to import a WSDL through Service Test Management, the ALM/QC extension for application components, open a connection to the ALM/QC sever. For details, see the "HP ALM/QC Connection Dialog Box" on page 208.

#### **2 Open the Import/ Select Service dialog box**

- To import a WSDL from the file system or ALM/QC, click  > **Import Service from File/ALM Application Component**. For details, see the "Select Service Dialog Box" on page 133.

---

**Note:** Before importing from Service Test Management, first connect to the ALM/QC server with Service Test Management (**File > ALM/QC Connection**).

---

- To import a WSDL from a Web site or UDDI repository, click



> **Import Service from URL/UDDI.**

- For a URL, select **Import from: URL**
- For a UDDI, select **Import from: UDDI**

For details, see the "Import Service from URL/UDDI Dialog Box" on page 134.

### 3 Select a source

- For **File System** or **ALM Application Components** (Quality Center) imports, click the appropriate button in the left pane. Navigate to the WSDL in the file system or the ALM/QC repository.
- For a **URL** import, click the Browse button adjacent to the **Address** field. This opens a browser window. Navigate to the WSDL and close the browser. Service Test automatically places the URL in the **Address** field.
- For a **UDDI**, click the Browse button adjacent to the **Address** field. This opens the Select Service from UDDI dialog box. Specify a UDDI address and select the service to import. For details, see the "Select Service from UDDI Dialog Box" on page 135.

#### 4 Set connection settings for URL/UDDI imports - optional

For URL or UDDI imports, if your WSDL must be accessed through a secure server or proxy machine, click **Connection Settings** to expand the dialog box. Enter the authentication information or the proxy server details.

#### 5 Complete the Import

Click **OK**. If Service Test succeeds in importing the WSDL, it displays it in the Toolbox palette under the **Web Services** branch.

### How to Set Security for a Web Service

This task describes how to configure security settings for Web Services.


This task includes the following steps:

- "Prerequisites" on page 119
- "Open the Security Setting view" on page 119
- "Set up a Web Services scenario - optional" on page 119
- "Set up a WCF Custom Binding scenario - optional" on page 120
- "Set up a WCF Federation scenario - optional" on page 120
- "Set up a WCF Http binding scenario - optional" on page 121
- "Configure advanced settings - optional" on page 122


## 1 Prerequisites

Import a Web Service and drag an operation into the canvas.

## 2 Open the Security Setting view

Click  in the Property Sheet to open the **Security Setting** view.


## 3 Set up a Web Services scenario - optional




- a In the **Service Details** dropdown list, select **Web Services** (default).
- b If relevant, open the **HTTP** tab and set the User Name, Password, Proxy URL (if not using the default proxy), Proxy User Name, and Proxy Password. These settings affect the transport level security. You can also set the **Keep alive** option and the **Connection timeout**, in seconds.
- c Click the **WS-Security** tab (optional).
  - ▶ To add a token, click  and select a token from the list, Provide the token details in the lower pane. The fields differ, based on the token type. For details, see "Message Level Security" on page 101.

---

**Note:** When adding a SAML token, if you have the complete SAML token string, you can paste it directly into the **AssertionIDReference** field. If you do not have the complete token string and you want to configure the token manually, make sure that the first row in the schema contains the **Assertion** property—not the **AssertionIDReference**. You can change this using the grid's dropdown menu.

---

- ▶ To add a message signature, (you must first add at least one token), click  and select which token to sign in the **Use token** box, usually an X.509 token. Provide the other required information: **Target token**, **Target path**, **Canonicalization algorithm**, **Transform algorithm**, **Inclusive namespaces list**, and **Signature options**.

- ▶ To add a message encryption, (you must first add at least one token), click  and select which token to sign in the **Use token** box. Provide the other required information: **Target token**, **Target path**, **Type**, **Key algorithm**, and **Session algorithm**.
- ▶ It is essential that you apply the security elements in the correct order. Use the **Up**  and **Down**  arrows to set the priorities of the security elements. The basic order is tokens, followed by message signatures, and then encryption. In addition, your service may also require a specific order for the tokens.
- d** Click the **WS-Addressing** tab (optional). Select the relevant version or **None** if WS-Addressing is not used.

For details about the security elements, see the "Web Service Security Scenario" on page 100. For user interface details see "Web Service Scenario Details" on page 84.

#### 4 Set up a WCF Custom Binding scenario - optional

To use WCF bindings:

- a** In the **Service Details** dropdown list, select **WCF Service (Custom Binding)**.
- b** In the **Transport** box, select a transport type: HTTP (default), HTTPS, TCP, or NamedPipe. Select an **Encoding** type: Text, MTOM, or WCF Binary.
- c** Configure the **Security**, **Net Security**, **Reliable Messaging**, and **Identities** information. For guidelines about setting these options, see "WCF Service (Custom Binding) Scenarios" on page 103.

For user interface details see "WCF Service (Custom Binding) Scenario Details" on page 88.

#### 5 Set up a WCF Federation scenario - optional

For details about this scenario type, see "WCF Service (Federation) Scenarios" on page 105.

- a** In the **Service Details** dropdown list, select **WCF Service (Federation)**.
- b** In the **Transport** box, select a transport type: HTTP (default) or HTTPS.



- c** Select an **Encoding** type: Text or MTOM.
- d** Select an **Authentication mode**. For the **Secure Conversation** mode, choose a **Bootstrap policy**.
- e** Set the server **Identities**—Click the Browse button to locate a certificate and optionally, specify the expected DNS.
- f** Provide the STS information. Click the **Browse** button in the Issuer Address box to open a browser. Navigate to address location.
- g** To apply referenced binding, click the Browse button in the **Referenced binding** field. In the Referenced Scenario dialog box, configure the referenced scenario.

For user interface details see "Security Settings View" on page 82.

## 6 Set up a WCF Http binding scenario - optional

- a** In the **Service Details** dropdown list, select **WCF Service (Http Binding)**.
- b** Select a **Client authentication type**: None, Windows, Certificate, or Username.
- c** For each authentication type, provide the required information. For details, see "WCF Services (Http Binding) Scenarios" on page 106.

For user interface details see "WCF Service (HTTP Binding) Scenario Details" on page 91.

## 7 Configure advanced settings - optional

For WCF type scenarios, click the **Advanced** button to set **Encoding**, **Advanced Standards** (such as Reliable Messaging), **Security** (such as Message Protection and token settings), and **HTTP/Proxy** preferences. For details, see "Advanced Settings Dialog Box" on page 137.

## 8 Save the scenario - optional

Your security scenario is automatically saved with the test. If, however, you also want to use the settings for another test, without having to redefine the scenario, you can save it.

In the Properties Sheet's Security view, click the **Save** button. Specify a location for the stss scenario file. To use the file in another test, click **Import**.

## How to Create a Simple Test for a REST Service

This task describes how to model a REST service and create a simple test. Optionally, you can create a template for each method, allowing you to reuse it within your test without having to provide the configuration settings each time. For details, see "How to Create a Template-Based REST Service" on page 125.

This task includes the following steps:

- "Add a new REST service" on page 123
- "Add a resource" on page 123
- "Add a method" on page 123
- "Create a test step" on page 123
- "Locate the endpoint URL" on page 123
- "Enter the input values" on page 123
- "Locate the request body" on page 124
- "Run the test and view the report" on page 124

## 1 Prerequisite

Study the structure of the REST Service body and determine which resources and methods you need to define.

## 2 Add a new REST service

In the Toolbox palette select the **REST Services** node. Choose **Add Service** from the shortcut menu. Provide a meaningful name for the service.

## 3 Add a resource

Select the service node and choose **Add Resource** from the shortcut menu. Provide a meaningful name for the resource. Repeat as required.

## 4 Add a method

Select the service or resource node, and choose **Add Method** from the shortcut menu. Provide a meaningful name for the method. Drag the method onto the canvas.

## 5 Create a test step

Drag the method you created above, into the **Test Flow** of the canvas. Expand the REST service step and click on the **HTTP Activity** box.

## 6 Locate the endpoint URL

Locate your REST service design document and copy the URL to the clipboard.

## 7 Enter the input values



- a** Open the Parameter view in the Property Sheet.
- b** Paste the clipboard contents, the endpoint URL, into the **Endpoint Address** row.
- c** Select the **HTTP Method**: GET, PUT, POST and so forth.

- d** Expand the **Request Header** node. Based on the operation details on the Help page, set the header name and value as follows: Type in the key and its value. For example, in the **Key** row you could type *Content-Type* and in the **Value** row, *text/xml*.

## 8 Locate the request body

- a** Return to the design document for the REST service and copy the Request body onto the clipboard.



- b** In the Property Sheet, open the **HTTP Activity** view and select a **Request Body** type from the dropdown list.

- To paste in plain text, select the **Text** type. For a request body of XML in the form of plain text, select **Text/xml**. The latter removes all extra spaces. Copy the Request body from the design document to the clipboard. Click within the **Body** section and press Ctrl-V to paste in the contents of the request. Modify the element values within the body as required.
- For the **XML** type, click **Import Schema** to load an XSD file. Edit the values for the schema in either the **Grid** or **Text** views
- For a **File** type, browse for the file with the Request body.
- For a **Post Form**, enter the keys and their values.

## 9 Run the test and view the report



Click the **Run** button or F5 to compile and run the test. After the test run, the Run Results Viewer opens.

In the left pane of the viewer window, select **Expand All** from the shortcut menu. Select the **HTTPActivity** node. In the **Captured Data** pane, verify that the **Response Body** values are correct.

## How to Create a Template-Based REST Service

This task describes how to set up a REST service template. You define the request details and parameters of the service directly from the toolbox, before dragging into a the Test Flow. Once you create the template, you can incorporate the REST activity in your test with minimal configurations.

This task includes the following steps:

- "Add a new REST service" on page 126
- "Add a resource" on page 126
- "Add a method" on page 126
- "Locate the endpoint URL" on page 126
- "Enter the REST input parameters - optional" on page 126
- "Define custom input parameters - optional" on page 127
- "Enter the Request body directly - optional" on page 127
- "Link the body request to a REST Input parameter - optional" on page 127

## 1 Prerequisite

Study the structure of the REST Service body and determine which resources and methods you need to define.

## 2 Add a new REST service

In the Toolbox palette select the **REST Services** node. Choose **Add Service** from the shortcut menu. Provide a meaningful name for the service.

## 3 Add a resource

Select the service node and choose **Add Resource** from the shortcut menu. Provide a meaningful name for the resource. Repeat as required.

## 4 Add a method

Select the service or resource node, and choose **Add Method** from the shortcut menu. Provide a meaningful name for the method.

## 5 Locate the endpoint URL

In your REST Service design document, determine the endpoint URL and copy it onto the clipboard.

## 6 Enter the REST input parameters - optional



- a** In the Toolbox, click on the method you created above. Open the **REST Input Parameter** view in the Property sheet.
- b** Paste the clipboard contents, the endpoint URL, into the **Endpoint Address** row.
- c** Select the **HTTP Method**: **GET**, **PUT**, **POST**, and so forth.
- d** Expand the **Request Header** node. Based on the operation details on the Help page, set the header name and value as follows: Type in the key and its value. For example, in the **Key** row you could type *Content-Type* and in the **Value** row, *text/xml*.

## 7 Define custom input parameters - optional

For methods that require input, such as **PUT** or **POST**, add the required input parameters. For methods that provide an output such as **GET**, add the required output parameters.



**a** In the Toolbox palette, select the method you created above. Open the Parameter view in the Property Sheet.



**b** Expand the Plus button and select **Add Input/Output Property**.

**c** Add custom parameters as required—provide a name and data type for the parameter.



**d** Manually provide values for the parameters or click the **Link to Source** button to map to existing data. For more information, see the "Select Link Source Dialog Box" on page 174.

## 8 Enter the Request body directly - optional

You can enter the Request body directly or data drive the values to use a query expression. To data drive the values, see the next step.

**a** Return to the design document for the REST service and copy the Request body onto the clipboard.



**b** In the Property Sheet, open the **HTTP Activity** view and click within the **Body** section. Press Ctrl-V to paste in the contents of the request. Make sure that there no extra spaces before or after the body text.

**c** Modify the element values within the XML body as required.

## 9 Link the body request to a REST Input parameter - optional

**a** Return to the design document for the REST service and copy the Request body onto the clipboard.

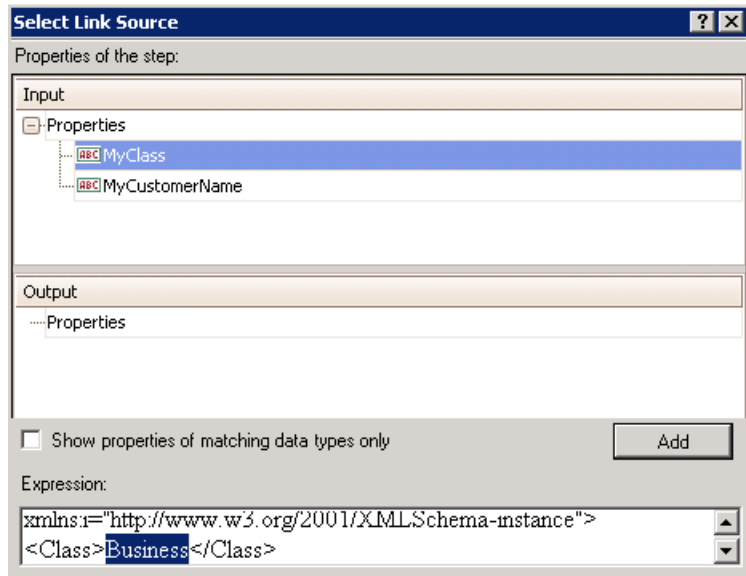


**b** In the Property Sheet, open the **HTTP Activity** view and click within the **Body** section.



**c** Click the **Link to Data** button to the right of the pane, to open the Select Link Source dialog box.

- d In the Select Link Source dialog box, check the **Select Multiple Sources** option. Paste the clipboard contents, the Request body, into the **Expression** box.
- e In the XML, highlight the value of the element you want to link. In the upper pane, double-click on the custom parameter you defined in step 7.



- f Repeat the same for all other elements that you want to data drive. Note the expression added in the Expression box in the bottom pane.

```
<Class>{Step..MyClass}</Class>
<CustomerName>{Step..MyCustomerName}</CustomerName>
```

- g Click **OK**. Service Test places the data in the HTTP body section.

You have now created a template for your REST method, complete with input parameters and the HTTP information. You can now drag the activity into the Test Flow and run it with minimal intervention.

For an example of creating a REST Service with data driving and data tables, see the *HP Service Test Tutorial* provided with the product.



## How to Set up Common Web Services Security Scenarios

This section illustrates several common security scenarios. The examples apply when using the default Web Service security scenario. For WCF-based services, the **WCF Service (Custom Binding)** scenario is recommended. For additional examples, see "Tips and Guidelines" on page 139.

### Authenticating with a Username Token

To send a message level username/password token (a **UserName** token):



**1** Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



**2** Click the **Add Token** button and add a **Username** token.

**3** Customize the token details, such as username and password.

### Signing with an X.509 Certificate

To send a message using a **X.509** certificate for a digital signature.



**1** Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



**2** Click the **Add Token** button and add an **X509 Certificate** token.

**3** Fill the token details to reference your private certificate. Make sure to enter a value in the **Logical Name** field.

**4** Select a **Reference type**. Since this token is used for a signature, the most common type is **BinarySecurityToken**.



**5** Click the **Add Message Signature** button. In the dropdown list, select the token you created in the previous steps.

---

**Note:** The certificate needs to be installed in the Windows certificate store. In the example above, you need to set the actual store name, store location, and subject name of your certificate.

---

## Signing a Specific Element with an X.509 Certificate

It is possible to sign only a specific element in a message. The following example signs a specific element using an XPATH expression:

**To send a message using a X.509 certificate for a digital signature.**



**1** Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



**2** Click the **Add Token** button and add an **X509 Certificate** token.

**3** Fill the token details to reference your private certificate. Make sure to enter a value in the **Logical Name** field.

**4** Select a **Reference type**. Since this token is used for a signature, the most common type is **BinarySecurityToken**.



**5** Click the **Add Message Signature** button. In the dropdown list, select the token you created in the previous step.

**6** In the **Target Path** field enter an XPATH expression, for example,  
`// *[local-name(.)='Body']`

## Encrypting with a Certificate

**To encrypt a message using the service's X.509 certificate:**



**1** Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



**2** Click the **Add Token** button and add an **X509 Certificate** token.

**3** Fill the token details to reference the server's public certificate. Make sure to enter a value in the **Logical Name** field.

**4** Since this token is used for encryption, use **Reference** as the **Reference type**.



**5** Click the **Add Message Encryption** button. In the dropdown list, select the token you created in the previous steps.

**6** To encrypt a specific message, specify an XPATH expression in the **Target Path** field.

## Authenticating with a Username Token and Encrypting with an X.509 Certificate

The following section describes how to send a **Username** token to the service and encrypt it with the server's **X.509** certificate:



**1** Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



**2** Click the **Add Token** button and add a **Username** token. Customize the token details, such as username and password.



**3** Click the **Add Token** button and add an **X509 Certificate** token.

**4** Fill the token details to reference the server's public certificate. Make sure to enter a value in the **Logical Name** field. Since this token is used for encryption, use **Reference** as the **Reference type**.



**5** Click the **Add Message Encryption** button. In the dropdown list, select the token you created in Step 3.

**6** To encrypt a specific message, specify an XPATH expression in the **Target Path** field.

## Encrypting and Signing a Message

This example shows how to sign a message using a private key and then encrypt it using the service's public key.



**1** Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



**2** Click the **Add Token** button and add an **X509 Certificate** token.

**3** Fill the token details to reference your private certificate. Make sure to enter a value in the **Logical Name** field. Select a **Reference type**. Since this token is used for a signature, use the **BinarySecurityToken** type.



**4** Click the **Add Token** button, and add another **X509 Certificate** token. Fill the token details to reference the server's public certificate. Make sure to enter a value in the **Logical Name** field. Since this token is used for encryption, use **Reference** as the **Reference type**.



**5** Click the **Add Message Signature** button. In the dropdown list, select the X509 token you created in Step 2.



- 6 Click the **Add Message Encryption** button. In the dropdown list, select the token you created in Step 4.

### Referencing an X.509 Certificate Using a Hash

In certain cases, you may be unable to reference a certificate with a subject name. This section shows how to reference the certificate using its unique hash.



- 1 Open the Security Settings view in the Property Sheet, select the **Web Services** scenario, and click the **WS-Security** tab.



- 2 Click the **Add Token** button and add an **X509 Certificate** token.
- 3 Fill in the token details. In the **Search Type** box, select **HashIdentifier**. Enter the certificate thumbprint (hash) in the **Search Value** field.
- 4 Add message encryption or a message signature as described in the previous sections.

---



---

## Reference

---




---

### Import Service User Interface

This section describes the user interface for importing a Web Service through a file system, URL, or Service Test Management.

### Select Service Dialog Box

Enables you to import WSDLs from a file system or Service Test Management, the ALM/QC extension for working with application components.

<b>To access</b>	
<b>Relevant tasks</b>	"Open the Import/ Select Service dialog box" on page 116

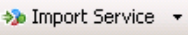
The following elements are included (unlabeled GUI elements are shown in angle brackets>):

UI Elements (A-Z)	Description
<b>File System</b>	Show WSDLs in the file system.
<b>ALM Application Components</b>	Show the WSDLs in the ALM/QC repository. <b>Note:</b> Only visible if there is an open connection to ALM/QC with a Service Test Management extension.

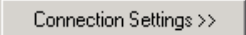

UI Elements (A-Z)	Description
<b>File name</b>	The contract name: <ul style="list-style-type: none"> <li>▶ <b>File System.</b> The file name of the WSDL.</li> <li>▶ <b>ALM Application Components.</b> The name of the application component or service.</li> </ul>
<b>Files of type</b>	The document type: <ul style="list-style-type: none"> <li>▶ <b>File System.</b> The contract file extension, such as <b>wSDL</b> or <b>All files</b>.</li> <li>▶ <b>ALM Application Components.</b> The type of application component—Web Service or JMS.</li> </ul>

## Import Service from URL/UDDI Dialog Box

Enables you to import WSDLs through a URL or UDDI.

<b>To access</b>	 > <b>Import Service from URL/UDDI.</b>
<b>Relevant tasks</b>	"Open the Import/ Select Service dialog box" on page 116
<b>See also</b>	"Select Service from UDDI Dialog Box" on page 135

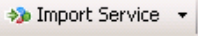

The following elements are included (unlabeled GUI elements are shown in angle brackets>):

UI Elements (A-Z)	Description
	Expands the dialog box to show the Authentication and Proxy settings.
	<b>Browse.</b> <ul style="list-style-type: none"> <li>▶ For <b>URL</b> imports: Opens a browser to allow you to navigate to a URL.</li> <li>▶ For <b>UDDI</b> Imports: Opens the Select Service from UDDI Dialog Box.</li> </ul>
<b>Address</b>	The URL or UDDI path of the WSDL.

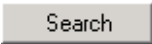
UI Elements (A-Z)	Description
<b>Use authentication settings</b>	The authentication settings by which to access the WSDL: <b>Username</b> and <b>Password</b> .
<b>Use proxy settings</b>	The authentication settings for the proxy server hosting the WSDL: <b>Server</b> , <b>Port</b> , <b>Username</b> , and <b>Password</b> .

## **Select Service from UDDI Dialog Box**

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

<b>To access</b>	 > <b>Import Service from URL/UDDI</b> . Select <b>Import from: UDDI</b> and click the  button.
<b>Relevant tasks</b>	"How to Import a WSDL-Based Web Service" on page 116
<b>See also</b>	"Import Service from URL/UDDI Dialog Box" on page 134

The following elements are included:

UI Elements (A-Z)	Description
	Retrieves the list of services that match the string in the <b>Service name</b> field. If no string is specified, it retrieves all available services.
<b>UDDI Address</b>	The name or IP address of the UDDI server inquiry API. <b>For Example:</b> http://lab1.devlab.ad:8090/uddi/inquiry.
<b>Version</b>	The UDDI version—2 or 3.
<b>Service Name</b>	A string in the service name by which to filter the list.
<list of services>	A list of the services that match the filter criteria. The grid shows the following information: <ul style="list-style-type: none"> <li>➤ <b>Service Name</b></li> <li>➤ <b>Description</b></li> <li>➤ <b>Service Contract</b></li> <li>➤ <b>Service ID</b></li> </ul>

## Security User Interface


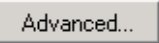
This section includes (in alphabetical order):

- [Select Service Dialog Box](#) on page 133
- [Import Service from URL/UDDI Dialog Box](#) on page 134
- [Select Service from UDDI Dialog Box](#) on page 135
- [Advanced Settings Dialog Box](#) on page 137
- [Select Certificate Dialog Box](#) on page 138



## Advanced Settings Dialog Box

This dialog box allows you to customize the security settings for your test.


<b>To access</b>	<ul style="list-style-type: none"> <li>➤ Select a Web Services step in the canvas</li> <li>➤ Open the Security Setting view in the Property Sheet.  .</li> <li>➤ Select the Custom Binding scenario.</li> <li>➤ Click  .</li> </ul>
<b>Important information</b>	For details, see "Advanced Security Settings" on page 109.
<b>Relevant tasks</b>	"How to Set Security for a Web Service" on page 118

User interface elements are described below:

UI Elements	Description
<b>Encoding tab</b>	Allows you to select the encoding type (Text, MTOM, or WCF Binary) and the WS-Addressing version. For details, see "Encoding" on page 109.
<b>Advanced Standards tab</b>	Allows you to configure the Reliable messaging and <b>Specify via address</b> settings. For details, see "Advanced Standards" on page 109.
<b>Security tab</b>	Allows you to enable secure sessions, negotiate service credentials, and set the protection level. From this tab you configure the token and derived key settings. For details, see "Security" on page 110.
<b>HTTP &amp; Proxy tab</b>	Allows you to configure HTTP details such as the transfer mode, response size, authentication scheme and realm, and proxy information. For details, see "HTTP and Proxy" on page 113.


## Select Certificate Dialog Box

This dialog box enables you to search and locate a certificate from a file or Windows store.

<b>To access</b>	<ul style="list-style-type: none"> <li>➤ Select a Web Service step in the canvas.</li> <li>➤ Open the Security Setting view in the Property Sheet.           <ul style="list-style-type: none"> <li> .</li> </ul> </li> <li>➤ Select a WCF type scenario in the <b>Service Details</b> box.</li> <li>➤ Click the Browse button adjacent to the <b>Server Certificate</b> or <b>Client Certificate</b> fields.</li> </ul>
<b>Relevant tasks</b>	"Set up a WCF Custom Binding scenario - optional" on page 120.

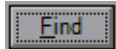
### Select Certificate from File

When you choose **Import from: File**, the dialog box shows the user interface elements are described below:

UI Elements	Description
	<b>Browse.</b> Allows you to locate the certificate file with a .cer or .pfx extension.
<b>File</b>	The complete path of the certificate file.
<b>Password (optional)</b>	The password required to access the certificate.

### Select Certificate from Windows Store

When you choose **Import from: Windows Store**, the dialog box shows the user interface elements are described below:

UI Elements	Description
	Begins the search for the certificate based on the Search text.
<b>Store location</b>	The store location, for example <b>Current User</b> .
<b>Store name</b>	The store name, for example, <b>AuthRoot</b> .

UI Elements	Description
<b>Search text</b>	The text to match in the certificate name. If left blank, the <b>Find</b> action retrieves all available certificates.
<certificate list>	A list of the certificates in the Windows store sorted by Subject, Issuer, Private, Store Location, and Store Name.
<b>Password (optional)</b>	The password required to access the certificate.

## Tips and Guidelines

This section provides a "how to" guide for testing WCF services and defining security scenarios.

### WCF Guidelines

This section describes how to use Service Test for testing WCF.

#### How do I test a WCF service?

Open the **Security View** in the Property Sheet and select a WCF scenario in the Scenario Details dropdown list.

If the Federation or WSHttpBinding scenarios are not appropriate, choose the **WCF Service (Custom Binding)** scenario, as it can all other bindings.

#### How do I test a WCF service that uses WSHttpBinding?

WSHttpBinding is one of the most popular bindings in WCF. In order to use this binding, select the **WCF Service (Http Binding)** scenario from the **Scenario Details** list.

In the Client Authentication box, select the client credential type that you use in your binding—Windows, Certificate, or Username. This value corresponds to the **MessageClientCredentialType** property of the WCF's WSHttpBinding.

**Windows** authentication is the most common value for a WCF services. If you are using the WCF default settings for your service, use this option. Other options are username, certificate, or none.

For some scenarios you should indicate whether to use the WCF proprietary negotiation mechanism to get the service credentials.

Use the Advanced scenario properties to control the usage of a secure session.

For details, see "Set up a WCF Http binding scenario - optional" on page 121.

### **How do I test a WCF service that uses CustomBinding?**

Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.

You can then customize many binding elements, such as your transport method, encoding, security, and reliable messaging.

For details, see "Set up a WCF Custom Binding scenario - optional" on page 120.

### **How do I test a WCF service that uses netTcp or namedPipe transport?**

Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.

Configure the transport to **TCP** or **NamedPipe**.

For details, see "Set up a WCF Custom Binding scenario - optional" on page 120.

### **How do I test a Federation scenario that uses an STS (Security Token Service)?**

For this scenario, you must define the communication properties for both the STS and the service. Use the built-in Federation scenario.

Open the **Security View** in the Property Sheet and select the **WCF Service (Federation)** scenario.

For this scenario, you must to define the communication properties for both the STS and the application server.

For details, see "Set up a WCF Federation scenario - optional" on page 120.

## How do I test a scenario that uses a WSE3 security configuration with a server certificate?

The following procedure describes how to set up a security scenario for WSE3.

- 1** Create a new test, import the WSDL for the W3ES service, and drag the operation into the canvas.
- 2** Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.
- 3** Set the Transport to **HTTP**, and the Encoding to **Text**.
- 4** Provide a username and password in the **Identities** section.
- 5** Click the Browse button adjacent to the Server Certificate field and specify the **Store Location**, **Store Name** and **Search text** (optional). Click **Find**, select the certificate, and click **Select**.
- 6** Provide the **Expected DNS**.
- 7** Click the Advanced button and configure the following settings:
  - a** **Encoding** tab— Encoding: Text, WS-Addressing: WSA 04/08 (for example).
  - b** **Security** tab—
    - **Enable secure session:** Enabled
    - **Negotiate service credentials:** Enabled
    - **Protection level:** Encrypt and Sign
    - **Message protection order:** Sign Before Encrypt
    - **Message security version:**  
WSSecurity11WSTrustFebruary2005WSSecureConversationFebruary2005 (first entry)
    - **Require Derived keys:** Enabled

For all other fields, use the default settings.

For details, see "Set up a WCF Custom Binding scenario - optional" on page 120.

## How do I test a scenario that uses mutual certificate authentication?

The following procedure describes how to set up a security scenario for mutual certificates and how to comply with a WSE3 security configuration.

- 1** Create a new test, import the WSDL for the service, and drag the operation into the canvas.
- 2** Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.
- 3** Set the Transport to **HTTP**, and the Encoding to **Text**.
- 4** Set the authentication mode to **MutualCertificate**.
- 5** In the **Identities** section, select server and client certificates. For details, see "Select Certificate Dialog Box" on page 138.
- 6** Provide the **Expected DNS**.
- 7** Click the **Advanced** button and configure the following settings:
  - a** **Encoding** tab— Encoding: Text, WS-Addressing: WSA 04/08 (for a WSE3 security configuration).
  - b** **Security** tab—**Require Derived keys**: Disabled

For all other fields, use the default settings.

For details, see "Set up a WCF Custom Binding scenario - optional" on page 120.

## How do I test a WCF service that uses the netTCP transport and authenticates using a client certificate on the transport level?

- 1** Create a new test, import the WSDL for the WCF service, and drag the operation into the canvas.
- 2** Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.
- 3** Set the Transport to **TCP**.
- 4** Set the Net Security to **SSL stream security**.
- 5** Open the Property Sheet's Event view.



- 6 Select the **BeforeApplyProtocolSettings** event. Click in the **Handler** column and select **Create a default handler** from the dropdown list.
- 7 Add the following definitions to the implementation section of the **SharedUserCode.cs** file.

```
var wcf = (HP.ST.Ext.WebServicesTechnologyRT.Models.WcfChannelBinding)args[1];
var ssl =
(HP.ST.Ext.WebServicesTechnologyRT.Models.WcfSslStreamSecurityChannel)wcf.Protocols.Channels[1];
ssl.RequireClientCertificate = true;
```

- 8 Save the test and run it.

### How do I configure a WCF binding with TCP transport to require an X509 client certificate?

The following procedure describes how to configure a WCF custom scenario to require an X509 client certificate in **nettcp**.

- 1 Create a new test, import the WSDL for the service, and drag the operation into the canvas.
- 2 Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.
- 3 Set the Transport to **TCP**.
- 4 Set the Net Security to **SSL stream security**.
- 5 Open the Property Sheet's Event view.
- 6 Select the **BeforeApplyProtocolSettings** event. Click in the **Handler** column and select **Create a default handler** from the dropdown list.
- 7 Add the following definitions to the implementation section of the **SharedUserCode.cs** file.

```
var wcf = (HP.ST.Ext.WebServicesTechnologyRT.Models.WcfChannelBinding)args[1];
var ssl =
(HP.ST.Ext.WebServicesTechnologyRT.Models.WcfSslStreamSecurityChannel)wcf.Protocols.Channels[1];
ssl.RequireClientCertificate = true;
```

For all other fields, use the default settings.

8 Save the test and run it.

## **General Security Testing**

This section provides a summary of using Service Test for general security testing.

### **How do I test a Web Service that uses SSL?**

Testing a secure site does not require any special configuration. If your service URL begins with **https**, SSL is automatically used.

If in addition to SSL you are using message-level security (for example a username) then you must configure the security for the message separately.

You can use the basic Web Services security scenario and specify the message-level security such as tokens and signatures.

You can also use the **WCF Service (Custom Binding)** scenario, or the **WCF Service (Http Binding)** scenario with the transport credentials.

### **How do I test a Web Service that require Windows authentication at the HTTP level?**

Open the **Security View** in the Property Sheet and select the **Web Services** scenario.

In the **HTTP** tab, specify the credentials.

For details, see "Set up a Web Services scenario - optional" on page 119.

### **How to I test a Web Service that uses WS-Security?**

Use the basic **Web Services** security scenario and open the **WS-Security** tab. Add the message-level security such as tokens, signatures, and encryption.

### **How do I configure the low-level details of my WS-Security tokens?**

In most cases, you can configure the low-level details as described in "Advanced Settings Dialog Box" on page 137.



## **Tips for Advanced Standards Testing**

This section provides guidelines for using Service Test in advanced standards testing.

### **How do I test a Web Service that uses MTOM?**

Open the **Security View** in the Property Sheet and select the **WCF Service (Custom Binding)** scenario.

Configure the **Encoding** to **MTOM**.

If your service requires advanced settings, click the **Advanced** button. For details, see the "Advanced Settings Dialog Box" on page 137.

For more information about the scenario, see "Set up a WCF Custom Binding scenario - optional" on page 120.

### **How do I change the WS-Addressing version of a service?**

Open the **Security View** in the Property Sheet and select the **Web Services** scenario.

Click the **WS Addressing** tab and select a version.

For details, see "Set up a Web Services scenario - optional" on page 119.

If your service uses WCF, use the appropriate scenario and configure the addressing version from the Advanced window's **Encoding** tab. For details, see the "Advanced Settings Dialog Box" on page 137.



# 4

---

## Data Handling

This chapter includes:

### Concepts

- ▶ [Data Handling Overview](#) on page 148
- ▶ [Defining Data](#) on page 148
- ▶ [Data Handling](#) on page 153

### Tasks

- ▶ [How to Create Data Queries](#) on page 158
- ▶ [How to Assign Data to Test Steps](#) on page 161
- ▶ [How to Set the Navigation Properties](#) on page 164
- ▶ [How to Data Drive a Test Step](#) on page 165

### Reference

- ▶ [Data Handling User Interface](#) on page 167

---

---

## Concepts

---

---

### Data Handling Overview

Before running a test, you may assign input and checkpoint data and specify how the data should be used.

Service Test's Data Window lets you view and maintain data for your test steps. You can import data from Excel files, create local data tables, or define XML queries as sources for your data. For details, see "How to Assign Data to Test Steps" on page 161.

You can data-drive the properties of a step. Data-driving is the mechanism by which Service Test automatically creates query expressions for the properties, that refer to data in a new, editable table. For details, see the "Data Driving Dialog Box" on page 167.

### Defining Data

Service Test allows you to assign data to your steps in several ways. You can set the data manually, link to existing data, or data drive the values and add data to tables creating a structure for the data for future use.

This section also includes:

- ▶ "Populating Data Manually" on page 149
- ▶ "Linking to a Data Query" on page 150
- ▶ "Data Driving" on page 151

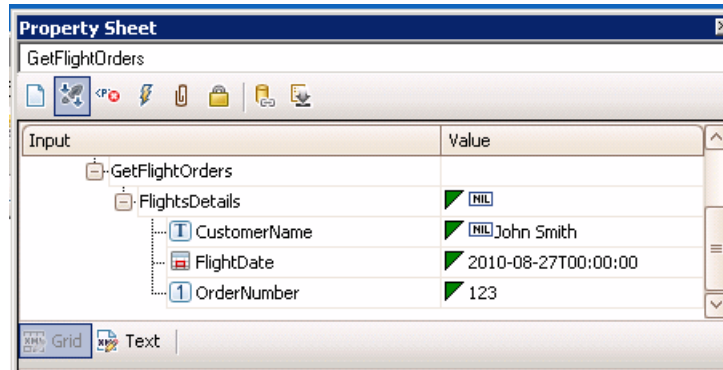
---

**Note:** For machines with Excel, the data tables follow standard Excel behavior—when you delete data from a table, the row remains without data. To remove a data row, select **Delete** from the row's shortcut menu. Otherwise, Service Test will run an iteration for that row, with empty values.

---

## Populating Data Manually

The Input/Checkpoints View grid allows you to view and edit the values of each input property. You can manually insert values into the grid for both simple properties and arrays.



You can also edit the values in the XML text mode. The editor also allows you to discard changes and return to the original using the **Revert** button.

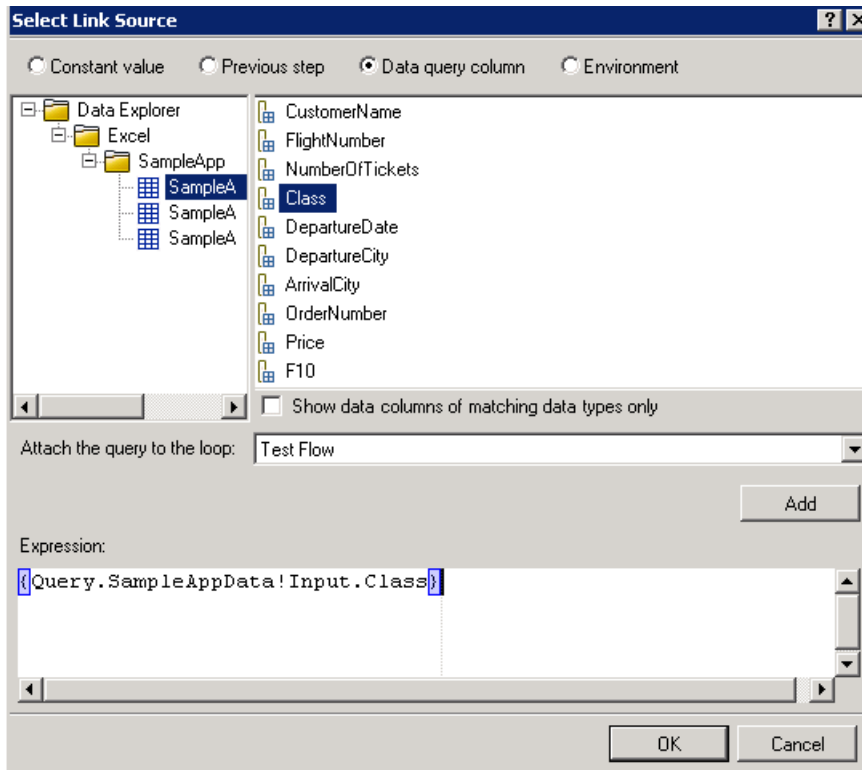
```
<?xml version="1.0" encoding="utf-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <GetFlightOrders xmlns="HP.SOAQ.SampleApp">
      <FlightsDetails>
        <CustomerName>John Smith</CustomerName>
        <FlightDate>2010-08-27T00:00:00+03:00</FlightDate>
        <OrderNumber>123</OrderNumber>
      </FlightsDetails>
    </GetFlightOrders>
  </Body>
</Envelope>
```

You can also populate the checkpoint values manually. In addition, after running the test at least once, you can load the replay values directly into the checkpoint using the **Load from Replay** button.

## Linking to a Data Query

The recommended way to define data for a step is by linking it to a data query. The Select Link Source dialog box allows you to link to data in several ways:

- ▶ Specify a constant value
- ▶ Reference the output of a previous step
- ▶ Use the data from a query in the Data window
- ▶ Use an test variable



You can also use a combination of these types: for example, a constant value, a data query column, and an test variable combined to make a single expression.

To use data query columns, you import an existing data file, such as an Excel spreadsheet or XML file. Alternatively, you can use a wizard to create a custom local table. For details, see the relevant section:

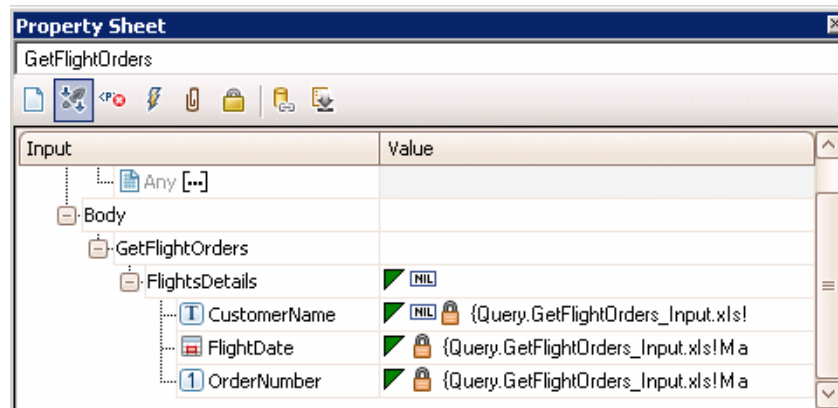
- "Add an Excel data query" on page 158
- "Add a local data table" on page 159
- "Add an XML data query" on page 159.

When preparing scripts for load testing, the Select Link Source dialog box also lets you select the way in which to retrieve the data and what to do when the data has run out. For details, see "Data Retrieval Options for Load Test Enabled Tests" on page 176.

For task details, see "How to Assign Data to Test Steps" on page 161.

## Data Driving

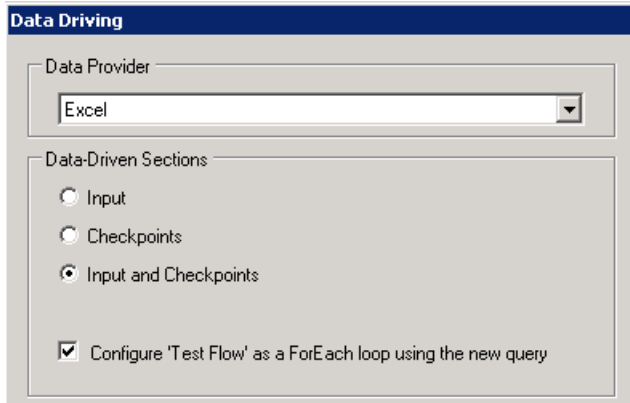
Data-driving is the mechanism by which Service Test automatically creates query expressions for properties. When you data drive a step, Service Test adds query expressions to the step's value fields.



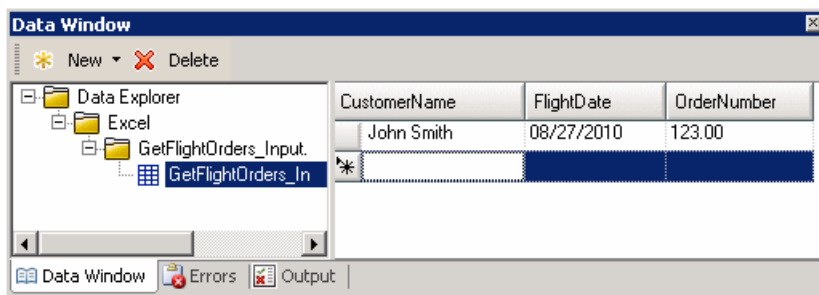
These queries refer to data in a new editable Excel table or XML structure, depending on the type of query you specified during the data driving.

You can customize data-driving in the following ways:

- ▶ Specify the type of data query to create—Excel or XML
- ▶ Indicate which properties to data drive— Input, Checkpoints, or both.
- ▶ Set the loop type. This overrides the regular Loop Settings.

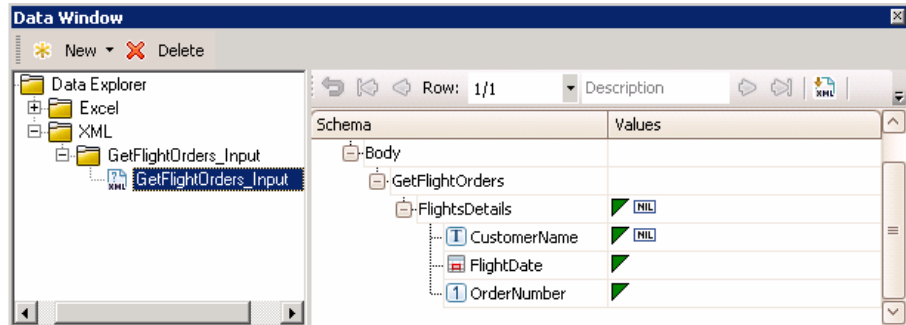


For Excel queries, you can manually add to and edit the table data in the Data Window.





For XML queries, you can manually edit the XML root.



For task details, see "How to Data Drive a Test Step" on page 165.

For user interface details, see "Data Driving Dialog Box" on page 167.

## Data Handling

Service test lets you indicate how to use the data query, how many times to loop through the data.

You can also define relations between data tables and queries.

This section also includes:

- ▶ "Navigating within the Data" on page 153
- ▶ "Child Relations" on page 155
- ▶ "Data Keywords" on page 156

### Navigating within the Data

Using the Data Navigation dialog box, you control the way you use the values in the data tables.

The Data Navigation settings work together with the loop settings described in "Loop Settings" on page 72.

If the loop is configured as a **ForEach** type, and you designate a specific query as the loop's collection, then the Data Navigation settings affect the number of iterations of the loop.

For queries that are not designated as the loop collection, but whose values are fetched by steps within the loop, the Data Navigation policy affects the data differently. It indicates the order in which the values are fetched from the data query, and assigned to steps within the loop.

The screenshot shows a dialog box titled "Data Navigation" with the following settings:

- Start:** Start at: Specific row; Row: 2
- Move:** Move by: 1 row(s); Forward
- End:** End at: Last row; Row: 7
- Upon reaching the last row:** Action: Wrap around

You can indicate the following:

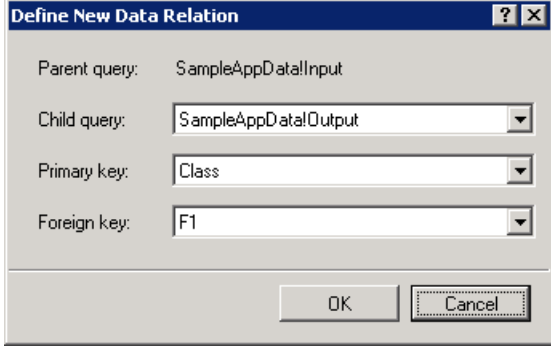
- ▶ **Start at.** At which row of data to begin. You can indicate a row number or a condition that must be met by the start row.
- ▶ **Move Direction.** The direction and the number of rows by which to advance and the condition that must be met. You can also move to a specific row or allow Service Test to use a random one.
- ▶ **End at.** At which row of data to end, and the condition that must be met by the start row. You can indicate a row number or a condition that must be met by the end row.
- ▶ **Last Row Behavior.**

- For **Forward** direction advancement, what to do when reaching the last row of data—continue using the last row for the rest of the iterations, or wrap around and begin again with the first row.
- For **Backward** direction advancement, what to do when reaching the first row of data—continue using the first row for the rest of the iterations, or wrap around and begin again with the last row.

For user interface details, see the "Data Navigation Dialog Box" on page 178.

## Child Relations

You can define data relations for all of your data tables. You can then use this relation as data for a test step.



The screenshot shows a dialog box titled "Define New Data Relation". It has a standard Windows-style title bar with a question mark icon and a close button (X). The dialog contains the following fields:

- Parent query: SampleAppData!Input
- Child query: SampleAppData!Output (selected from a dropdown menu)
- Primary key: Class (selected from a dropdown menu)
- Foreign key: F1 (selected from a dropdown menu)

At the bottom of the dialog are two buttons: "OK" and "Cancel".

To create a relation, you select a parent query and specify the following:

- **Child Query.** The query containing the column whose data you want to use in the relation.
- **Primary key.** The column in the parent query used as a primary key.
- **Foreign key.** The column in the child query used as a foreign key.

For task details, see "Create a new child relation - optional" on page 161.

 **Data Keywords**

You can use keywords to customize the test run and validation. To use a keyword, type it into the **Value** column or table row. Enclose the keyword with the # character. The keyword names are not case-sensitive.

Service Test provides keywords for both Input properties and checkpoints.

**Input Keywords**

For input properties, you can enter a keyword into one or more data provider sources, such as a data table or XML code.

Keyword	Description
<b>#SKIP#</b>	Omits this element from the XML of the request.
<b>#NIL#</b>	Adds a <b>nil=true</b> attribute to the property's XML in the request. For example, <code>&lt;name John Doe nil="true"&gt;&lt;/name&gt;</code> .  If the XML element is not nillable, Service Test reports this to the log.

## Checkpoint Keywords

You can use keywords for checkpoints, by entering the keyword into the data table or XML associated with the checkpoint. Service Test provides the following keywords for checkpoints.

Keyword	Description
<b>#SKIP#</b>	The checkpoint with the #SKIP# value will not be validated.
<b>#NIL#</b>	During the test run, Service Test checks that the element has a <b>nil="true"</b> attribute in its response.
<b>#NOT_FOUND#</b>	Indicates that the element should not be found in the XML of the response. This setting ignores the comparison operator—it only checks for the absence of a value.
<b>#EXISTS#</b>	Indicates that the element should be found in the XML of the response. This setting ignores the comparison operator—it only checks for the presence of a value.

---

---

## Tasks

---

---

### How to Create Data Queries

This task describes how to define data queries before selecting a test step. After creating a test step, you link it to an existing data query.

This task includes the following steps:

- "Add an Excel data query" on page 158
- "Add a local data table" on page 159
- "Add an XML data query" on page 159
- "Create a new child relation - optional" on page 161

#### Add an Excel data query

- 1** Make sure the Data Window is visible. If not, select **View > Data Window**.
- 2** In the Data Window pane, select **New > Excel File**.
- 3** In the Add New Excel File Data Query dialog box, browse for the file and indicate if the first row is a header row.
- 4** Specify a query name or accept the default one.
- 5** Select an import mode: **Referenced** or **Embedded**. The Data Explorer shows the table in the right pane.
- 6** Click **OK**.
- 7** In the Data Explorer, select the query's node and move within the table using the keyboard arrows. Click the grid cells to manually set values.

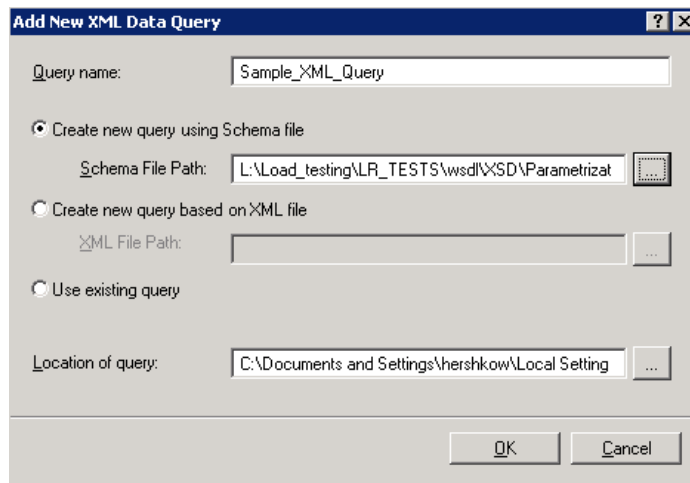
For user interface details, see the "Add New Excel File Data Query Dialog Box" on page 168.

## Add a local data table

- 1 Show the Data Window. Select **View > Data Window**.
- 2 In the Data Window pane, select **New > Local Table**.
- 3 Click **Add** to create a column for the data query.
- 4 In the Add Columns dialog box, specify a column name and description. Select a data type from the dropdown list. Click **OK**.
- 5 Repeat steps **c** and **d** for each column you want to create.
- 6 Use the arrows to reorder the columns as required.
- 7 Click **Next** to specify a query name. Click **Finish**.
- 8 In the Data Explorer, select the table's node and move within the table using the keyboard arrows. Click the grid cells to manually set values.

## Add an XML data query

- 1 Show the Data Window. Select **View > Data Window**.
- 2 In the Data Window pane, select **New > XML**. The Add New XML Data Query dialog box opens. For user interface details, see the "Add New XML Data Query Dialog Box" on page 170.



- 3 Provide a **Query name**.

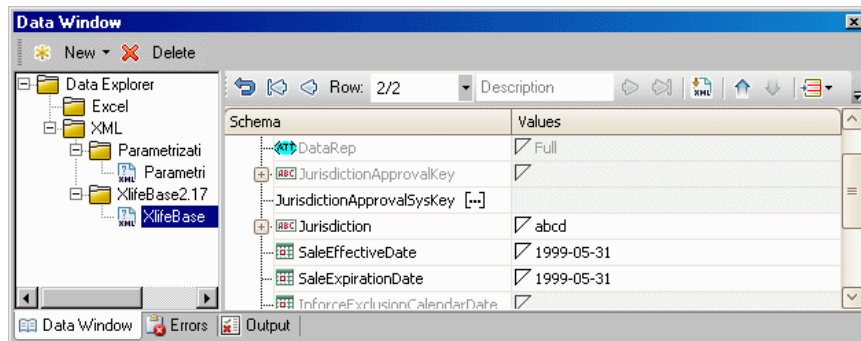
- 4 Select a source or schema option:
  - ▶ **Create new query using schema file.** Browse for the **Schema File Path**.
  - ▶ **Create new query based on XML file.** Browse for the **XML File Path** representing the structure of the data. You can add data at a later time.
  - ▶ **Use existing query.** Browse for the **Location of Query**. Queries are stored as subfolders of the script's **DataQueries** folder. Make sure to use the query name as it appears in the folder, without the suffix. For example, for the file **Query1\_DataSource.xml** file, enter *Query1* in the **Query name** field.

---

**Tip:** Choose **Test > Open Script Directory** to view the files.

---

- 5 Click **OK**.
- 6 Expand the query's node in the Data Window.



- 7 Edit the XML data in the grid. Add rows as described in the "XML Query Grid - Toolbar" on page 171.
- 8 To load XML values, click the **Load data from an XML file** button.





## Create a new child relation - optional

This step lets you add child relations using primary and foreign keys. This applies to Excel type queries.

- 1 Add at least two Excel files or one file with two worksheets, containing the foreign and primary keys.
- 2 In the Data Window, select the parent table in the Data Explorer tree.
- 3 Click in the Property Sheet to open the **Data Query Properties** view.
- 4 Click the **Add** button. The Define New Data Relation dialog box opens.
- 5 Specify the details of the new relation. For user interface details, see "Define/Edit New Data Relation Dialog Box" on page 181.
- 6 To edit a data relation, double-click the entry or click **Edit**.
- 7 To delete a data relation, select the entry and click **Remove**.



## How to Assign Data to Test Steps

This task describes how to link the test step properties to data queries through the following steps:

- "Open the Select Link Source dialog box" on page 161
- "Select a linking option" on page 162
- "Assign array data - no leaf nodes" on page 163

### 1 Open the Select Link Source dialog box



- a Display the step's properties. Select **View > Property Sheet** and open the **Input/Checkpoints** view.



- b Click the **Link Source** button in the right corner of the row.

## 2 Select a linking option

- a** In the Select Link Source dialog box, select a data query option:
    - **Constant value.** Enter the value manually.
    - **Previous step.** Select the step in the left pane and its property in the right pane.
    - **Test variable.** Select a user or system variable. For information on defining user variables and profiles, see "Define user variables" on page 17.
    - **Data query column.** For properties with simple data structures or with both leaf nodes and arrays—in the left pane, select a query. In the right pane, double-click on the relevant node. For properties that only have arrays, see step 3 on page 163.
  - b** To use a combination of several data sources for one parameter, click **More** to expand the Select Link Source dialog box. Select the data sources one-by-one and click **Add**. If necessary, edit the expression manually.
  - c** Click **OK** to complete the linking of the data. Service Test copies the values or expressions into the **Value** column of the grid.
- 

### Note:

- Linking data queries to properties only applies to steps within a loop.
  - When linking data queries to an array with multiple elements, you only link the data to the first array element. Service Test will apply the data to the next elements according to the availability of the data. For example, if the Excel table contains three rows of data for the current parent query row, it will populate three array elements.
  - Linking a leaf node to a complex XML node is only supported for string type data.
- 

For user interface details, see the "Select Link Source Dialog Box" on page 174. To

### 3 Assign array data - no leaf nodes

Follow these steps if the property only contains an array, and there are no leaf nodes outside the array.

- a** Create a data relation (if required) as described in "Create a new child relation - optional" on page 161.
- b** Click within the frame of Test Flow, or the relevant loop, and open the **Data Queries** view in the Property Sheet.
- c** Click the **Add** button. The Attach Query to Loop dialog box opens.
- d** Select the parent query from the list. Click **OK**.
- e** To edit the data navigation properties, click **Edit**. Set the preferences and click **OK**.
- f** Select the step in the canvas. In the Property Sheet's **Input/Parameter's** view, click within the row of the array element you want to assign.
- g** Click the **Link Source** button in the right corner of the row. Select the **Data query column** option and select the desired node.



## How to Set the Navigation Properties

This task describes how to set the navigation preferences for data stored in tables. You can set different navigation properties for each data query. For details, see "Navigating within the Data" on page 153.

### 1 Open the Data Queries view



In the canvas, click within the loop frame, but not within a test step. In the Property Sheet, click the **Data Queries** view button (only visible for steps within a loop).

### 2 Select a query

Click **Add** to select a query. The Attach Query to Loop dialog box opens. Double-click a query.

### 3 Open the Data Navigation dialog box

in the **Data Queries** view, select the query with your data. Click **Edit** to open the Data Navigation dialog box.

### 4 Set the navigation properties

- a** In the Data Navigation dialog box, specify the **Start** and **End** rows.
- b** Indicate the direction in which to move when retrieving data from the table: **Forward** or **Backward**, and the number of rows by which to advance for each iteration.
- c** Specify the action to do when reaching the last row- **Wrap around** or **Keep using that row**.
- d** Click **OK**.

For user interface details, see the "Data Navigation Dialog Box" on page 178.

## How to Data Drive a Test Step

This task describes how to data drive a test step. Data driving is Service Test's mechanism for automatically creating a query expressions for step properties.

For further details, see "Data Driving" on page 151.

This task includes the following steps:

- "Prerequisites" on page 165
- "Open the Input Properties view" on page 165
- "Include the data" on page 165
- "Data drive the test step" on page 165
- "Enter values" on page 166

### 1 Prerequisites

Create a step in the canvas. For details, see "Create the test flow" on page 42.

### 2 Open the Input Properties view

Display the step's properties grid. Select **View > Property Sheet** and open the **Input/Checkpoints** view.



### 3 Include the data

To include optional properties in the data driving, toggle the triangle icon adjacent to the property. A filled-in triangle indicates an included property.

### 4 Data drive the test step



- a** Click the **Data Drive** button. The Data Driving dialog box opens.
- b** Choose a Data provider: **Excel** or **XML**.
- c** Select the properties upon which you want to apply data driving: input properties, checkpoints, or both.

- d** Indicate whether you want to **Configure 'Test Flow' as a For Each loop using the new query**. This option repeats the test steps according to the number of data rows in the Excel or XML data query. If you disable this option, you can manually set the number of iterations through the loop properties. This setting overrides the general loop properties. For details, see "Loop Settings" on page 72.
- e** Click **OK**.
- f** Service Test prompts you to confirm the action. Click **OK**.

Service Test populates the value column with queries. The queries are preceded by informational icons, indicating read-only status or unmatching data types.

---

**Note:** If your checkpoints are within an array, you must create an array element before data driving. Expand the **Body** node, select the array and choose **Add Array Element** from the shortcut menu.

---

For user interface details, see the "Data Driving Dialog Box" on page 167.

## 5 Enter values

In the Data Window (**View > Data Window**), edit the columns of the newly created data table(s) or XML source.

---



---

## Reference

---



---


### Data Handling User Interface

This section includes (in alphabetical order):

- ▶ [Data Driving Dialog Box](#) on page 167
- ▶ [Add New Excel File Data Query Dialog Box](#) on page 168
- ▶ [Add New Local Table Data Query Dialog Box](#) on page 169
- ▶ [Add New XML Data Query Dialog Box](#) on page 170
- ▶ [Attach Query to Loop Dialog Box](#) on page 173
- ▶ [Select Link Source Dialog Box](#) on page 174
- ▶ [Data Navigation Dialog Box](#) on page 178
- ▶ [Define/Edit New Data Relation Dialog Box](#) on page 181

### Data Driving Dialog Box

Enables you to populate the input property and checkpoint values for a step, with data query statements.

<b>To Access</b>	
<b>Relevant Tasks</b>	"How to Data Drive a Test Step" on page 165
<b>See Also</b>	"Add New Excel File Data Query Dialog Box" on page 168

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


UI Elements	Description
<b>Data Provider</b>	The source type of the data: <b>Excel</b> or <b>XML</b> .
<b>Data Driven Sections</b>	The section upon which to apply the data driving: <ul style="list-style-type: none"> <li>➤ <b>Input.</b> All input properties listed in the grid.</li> <li>➤ <b>Checkpoints.</b> All checkpoints listed in the grid. For array type Output properties, you must create at least one array element in order to enable data driving.</li> <li>➤ <b>Inputs and Checkpoints.</b> All entities in the grid.</li> </ul>
<b>Configure 'Test Flow/ Loop' in a ForEach loop using the new query</b>	Runs the steps in the Test Flow or Loop using a <b>For Each</b> type loop, with the values from the newly created query. It performs one iteration for each row in the query. <b>Note:</b> This option overrides the configuration that you may have set for the Test Flow/Loop settings. For details, see "Loop Settings" on page 72.

## Add New Excel File Data Query Dialog Box

Enables you to add new Excel queries to the Data Explorer within the Data tab.

<b>To Access</b>	<b>View &gt; Data Window</b> and select <b>New &gt; Excel File</b> .
<b>Relevant Tasks</b>	"Add an Excel data query" on page 158

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

UI Elements	Description
	<b>Browse.</b> Enables you to locate the Excel file containing the query data.
<b>Excel file path</b>	The complete path of the Excel file containing the data.



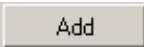
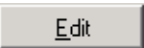
UI Elements	Description
<b>Excel file contains header row</b>	Indicates whether the data in the Excel table contains a header row. Header rows are ignored when running the test.
<b>Query name</b>	The name of query as it will appear in the Data Window's Excel file node. <b>Default:</b> The name of the file.
<b>Mode</b>	The mode of storage for the query: <ul style="list-style-type: none"> <li>▶ <b>Referenced Query.</b> Links to the Excel file at its original location. If the data changes at its source, it will affect your test. In addition, if you modify the data in Service Test, it will affect the data at its source.</li> <li>▶ <b>Embedded Query.</b> Copies the Excel file locally. If the data changes at its source, it will not affect your test.</li> </ul>



## Add New Local Table Data Query Dialog Box

Enables you to create local tables in the Data window for use with your test.

<b>To Access</b>	<b>View &gt; Data Window</b> and select <b>New &gt; Local Table</b> .
<b>Relevant Tasks</b>	"Add a local data table" on page 159
<b>See Also</b>	<ul style="list-style-type: none"> <li>▶ "Add New Excel File Data Query Dialog Box" on page 168</li> <li>▶ "Add New XML Data Query Dialog Box" on page 170</li> </ul>

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

UI Elements	Description
	Opens the <b>Add Columns</b> dialog box for providing a new column's properties.
	Opens the <b>Edit Columns</b> dialog box for modifying the properties of the selected column.

UI Elements	Description
	Deletes the selected data table column.
	Raises/Lowers the selection in the column list.
<b>Columns</b>	A list of the columns in the local data table.

### Add/Edit Columns Dialog Box

The Add/Edit Columns dialog box lets you define the properties for a new column or edit the properties of an existing one.

User interface elements are described below:

UI Elements (A-Z)	Description
<b>Column Name</b>	An arbitrary name for the column.
<b>Data Type</b>	The type of data in the column: <ul style="list-style-type: none"> <li>▶ String</li> <li>▶ Integer</li> <li>▶ Double</li> <li>▶ Date</li> </ul>
<b>Description</b>	A description of the data in the column.

### Add New XML Data Query Dialog Box

Enables you to add new XML queries to the Data Explorer within the Data tab.





<b>To Access</b>	<b>View &gt; Data Window</b> and select <b>New &gt; XML</b> .
------------------	---








<b>Relevant Tasks</b>	"Add an XML data query" on page 159
<b>See Also</b>	<ul style="list-style-type: none"> <li>▶ "Add New Excel File Data Query Dialog Box" on page 168</li> <li>▶ "Add New Local Table Data Query Dialog Box" on page 169</li> </ul>

The following elements are included:

UI Elements	Description
<b>Query name</b>	The display name of the query.
<b>Create new query using Schema file</b>	Loads the specified schema into the Data Explorer window as a basis for the new query.
<b>Create new query based on XML file</b>	Creates a new schema based on the XML file specified in the <b>XML file path</b> box.
<b>Use existing query</b>	Uses an existing query specified in the <b>Location of query</b> box. Since this query is referenced, changes made to the query will be reflected in all tests using the query.


### XML Query Grid - Toolbar

UI Elements (from left to right)	Description
	<b>Go back.</b> Go back to previous view.
	<b>Go to start.</b> Go to the first row.
	<b>Go to previous row.</b> Go to the previous row (one row up).
<b>Row</b>	The current row number out of the total rows, such as 1/3 and so forth.
<b>Description</b>	A title for the row currently displayed.
	<b>Go to next.</b> Go to the next row (one row down).

UI Elements (from left to right)	Description
	<b>Go to last.</b> Go to the last row.
	<b>Load XML.</b> Loads data from an XML file into the current row.
	<b>Move up.</b> Moves the current row up one position.
	<b>Move down.</b> Moves current row down one position.
	Adds a new empty row at one of the following locations. <ul style="list-style-type: none"> <li>▶ <b>Add after current</b></li> <li>▶ <b>Add at start</b></li> <li>▶ <b>Add at end</b></li> </ul>
	Duplicates the current row at one of the following locations. <ul style="list-style-type: none"> <li>▶ <b>Duplicate after current</b></li> <li>▶ <b>Duplicate at start</b></li> <li>▶ <b>Duplicate at end</b></li> </ul>
	Deletes the current row.

## Attach Query to Loop Dialog Box

This dialog box enables you to add a data query to the current loop.


<b>To access</b>	<p>Follow these steps:</p> <ul style="list-style-type: none"> <li>▶ In the canvas, click in a loop, but not within a step.</li> <li>▶ In the Property Sheet, select the <b>Data Query</b> view. .</li> <li>▶ Click <b>Add</b>.</li> </ul>
<b>Important</b>	<p>You must first define data in the Data Window in order to generate a list of queries.</p> <p>When you link any step in the loop to the data, it automatically adds a query to the loop.</p> <p>For details, see "How to Assign Data to Test Steps" on page 161.</p>
<b>Relevant tasks</b>	"How to Set the Navigation Properties" on page 164
<b>See also</b>	"Data Navigation Dialog Box" on page 178

User interface elements are described below:


Element	Description
<b>Select a data query</b>	A list of the available data queries.

## Select Link Source Dialog Box

This dialog box enables you to select a data source for the step's properties.

<b>To access</b>	Click  in the <b>Value</b> column of the property for which you want to select a data query.
<b>Important information</b>	<ul style="list-style-type: none"> <li>▶ To see the icon, click in the property row and move the cursor to the right of the cell.</li> <li>▶ A step property not contained within a loop, cannot be linked to a data query. For example, the For Loop's <b>Number of Iterations</b> property, cannot be linked to a data query, since it is not contained within the Test Flow loop.</li> </ul>
<b>Relevant tasks</b>	"How to Assign Data to Test Steps" on page 161
<b>See also</b>	<ul style="list-style-type: none"> <li>▶ "How to Create Data Queries" on page 158</li> <li>▶ "How to Set the Navigation Properties" on page 164</li> </ul>

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

UI Elements (A-Z)	Description
	Expands the dialog box to allow you to use multiple data sources for an expression, and edit the query expression manually.
<data source>	<p>The source for the property value:</p> <ul style="list-style-type: none"> <li>▶ <b>Constant value.</b> A constant value to use for all iterations.</li> <li>▶ <b>Previous step.</b> A property value from a previous step.</li> <li>▶ <b>Data query column.</b> A value from a data query—Excel, XML, or local table.</li> <li>▶ <b>Test Variables.</b> A value of an test variable—either user or system.</li> </ul>
<preceding step list>	<p>A list of the steps preceding the current one. Select the step whose output value you want to use as input.</p> <p><b>Note:</b> Visible only when choosing <b>Previous step</b>.</p>

UI Elements (A-Z)	Description
<step properties grid>	A schema view of the selected step's properties. <b>Note:</b> Visible only when choosing <b>Previous step</b> .
<data queries and their columns>	A list of the Excel, XML, or local table queries in the left pane, and their columns or schema (for XML) in the right pane. <b>Note:</b> Visible only when choosing the <b>Data query column</b> option.
<user variable grid>	A list of the user-defined test variables and their values in the current profile. <b>Note:</b> Visible only when choosing the <b>Test Variables</b> option.
<system variable grid>	A list of the system environment variables and their values: ControllerHostName, IsLoadTest, ScenarioID, SystemTempDir, GroupName, TestDir, TestName, LocalHostName, OS, OSVersion, ProductDir, ProductName, ProductVer, and UserName. <b>Note:</b> Visible only when choosing the <b>Test Variables</b> option.
Enter a value	The constant value to use for this property in all iterations. <b>Note:</b> Visible only when choosing <b>Constant value</b> .
Show data columns of matching data types only	Hides all columns (or schema rows for XML) whose data types do not match the input property whose value you want to set. <b>Note:</b> Visible only when choosing the <b>Data query column</b> option.

UI Elements (A-Z)	Description
<b>Show properties of matching data types only</b>	Hides all properties whose data types do not match the one whose value you want to set. <b>Note:</b> Visible only when choosing the <b>Previous step</b> .
<b>Attach the query to the loop</b>	A dropdown list of the loops in the test containing the activity. The query is only associated with the selected loop. <b>Note:</b> Visible only when choosing the <b>Data query column</b> option.

### Data Retrieval Options for Load Test Enabled Tests

For Load Test Enabled tests, the Select Link Source dialog box provides additional controls for the Input properties, when choosing the **Data query column** option.

User interface elements are described below:


UI Elements	Description
<b>Attach the query to the loop</b>	The loop with which to associate the query (for future implementations).



UI Elements	Description
<p><b>Use a unique value for each Virtual User when load testing</b></p>	<p>Assigns a unique value to each Virtual User, from the data table, with the following properties:</p> <ul style="list-style-type: none"> <li>▶ <b>Query parameter.</b> The query parameter to use as a source of values for the selected input property.</li> <li>▶ <b>Update value on.</b> The frequency by which LoadRunner will update the values: <b>Each Iteration</b>, <b>Each Occurrence</b> (of the query parameter), or <b>Once</b>.</li> <li>▶ <b>When out of values.</b> The action to do when reaching the end of the data table: <ul style="list-style-type: none"> <li>▶ <b>Abort Vuser.</b> End the test run for the Virtual User.</li> <li>▶ <b>Continue Cyclic.</b> Return to the top of the data table and use the data in a cyclic manner.</li> <li>▶ <b>Continue with Last.</b> Use the last value retrieved from the data table for all subsequent occurrences.</li> </ul> </li> </ul>
<p><b>Allocate Virtual User values in the Controller</b></p>	<p>Allocates a specific amount of values from the data table for the Virtual User:</p> <ul style="list-style-type: none"> <li>▶ <b>Automatically allocate block size.</b> Automatically allocates a block size from the data based on the number of iterations and Virtual Users (only for Update value on <b>Each Iteration</b>).</li> <li>▶ <b>Allocate &lt;amount&gt; values for each Virtual User.</b> Allocates a specific number of values for each Virtual User.</li> </ul> <p><b>Note:</b> Allocation is only possible when choosing Update value on <b>Each Occurrence</b> or <b>Each Iteration</b>. For details, see the <i>HP LoadRunner Controller User Guide</i>.</p>

## Data Navigation Dialog Box

This dialog box enables you to set the data navigation properties for your test loop. You can set the direction in which to use the data, from where to begin, and the condition for selecting data.

<b>To access</b>	<p>Follow these steps:</p> <ol style="list-style-type: none"> <li><b>1</b> In the canvas, click inside the loop frame—not on a test step.</li> <li><b>2</b> In the Property Sheet, select the <b>Data Query</b> view .</li> <li><b>3</b> Click <b>Add</b>. The Attach Query to Loop dialog box opens.</li> <li><b>4</b> Select a query to attach and click <b>OK</b>. The query is added to the grid.</li> <li><b>5</b> Click <b>Edit</b>.</li> </ol>
<b>Relevant tasks</b>	"How to Set the Navigation Properties" on page 164
<b>See Also</b>	"Navigating within the Data" on page 153


User interface elements are described below:

GUI Element	Description
<b>Start at</b>	<p>The row within the query from which to begin.</p> <ul style="list-style-type: none"> <li>▶ <b>Start at.</b> The row of data from which to begin: <b>First row, Last row, Specific row, Random row, or First row matching.</b></li> <li>▶ <b>Row.</b> The row number from which to begin (only available when selecting <b>Specific row</b>).</li> <li>▶ <b>Condition.</b> The condition the data row must meet in order to start the looping (only available when selecting the <b>First row matching</b> option): <ul style="list-style-type: none"> <li>▶ <b>Column.</b> One of the query's columns.</li> <li>▶ <b>Comparison operator.</b> =, !=, &gt;, &gt;=, &lt;, &lt;=, Starts, Ends, Contains, or Regex (depending on row's data type).</li> <li>▶ <b>Value.</b> The value to be matched in order to use this row of data.</li> </ul> </li> </ul>
<b>Move</b>	<p>The direction and amount of rows by which to advance within the data.</p> <ul style="list-style-type: none"> <li>▶ <b>Move by.</b> The number of rows to advance.</li> <li>▶ <b>Direction.</b> The direction in which to move: <b>Forward</b> or <b>Backward.</b></li> </ul>

GUI Element	Description
<p><b>End at</b></p>	<p>When to stop the looping through the data: <b>First row, Last row, Specific row, or First row matching</b></p> <ul style="list-style-type: none"> <li>➤ <b>Row.</b> The row number in which to end (only available when selecting <b>Specific row</b>).</li> <li>➤ <b>Condition.</b> The condition the data row must meet in order to end the looping (only available when selecting the <b>First row matching</b> option):                             <ul style="list-style-type: none"> <li>➤ <b>Column.</b> One of the query's columns.</li> <li>➤ <b>Comparison operator.</b> =, !=, &gt;, &gt;=, &lt;, &lt;=, Starts, Ends, Contains, or Regex (depending on row's data type).</li> <li>➤ <b>Value.</b> The value to be matched in order to use this row of data.</li> </ul> </li> </ul> <p><b>Note:</b> This setting stops the looping only when the loop type is set to '<b>For Each</b>' (see "Loop Settings" on page 72) and the current query whose navigation settings you are editing, is the same query that is linked to the loop.</p>
<p><b>Upon reaching the last row - Action:</b> (Forward direction)</p>	<p>The action to take when reaching the last row of data.</p> <ul style="list-style-type: none"> <li>➤ <b>Keep using that row.</b> Keep using the last row of data for all subsequent loops.</li> <li>➤ <b>Wrap around.</b> Start again from the first row of data.</li> </ul>
<p><b>Upon reaching the first row - Action:</b> (Backward direction)</p>	<p>The action to take when reaching the first row of data.</p> <ul style="list-style-type: none"> <li>➤ <b>Keep using that row.</b> Keep using the first row of data for all subsequent loops.</li> <li>➤ <b>Wrap around.</b> Start again from the last row of data.</li> </ul>

## Define/Edit New Data Relation Dialog Box

This dialog box enables you to define a new data relation for a query.

<b>To access</b>	<p><b>Follow these steps:</b></p> <ol style="list-style-type: none"> <li><b>1</b> Add a data query through the Data Window and select its node in the left pane.</li> <li><b>2</b> Click in the Property Sheet pane. The <b>Data Query Properties</b>  view opens.</li> <li><b>3</b> Click <b>Add</b> or <b>Edit</b>.</li> </ol>
<b>Relevant tasks</b>	"Create a new child relation - optional" on page 161

User interface elements are described below:

GUI Element	Description
<b>Child query</b>	The sheet or table to use for the query.
<b>Primary key</b>	A column in the parent query to use as a primary key for the child relation.
<b>Foreign key</b>	The column in the child query to use as a foreign key for the child relation.



# 5

---

## Running Tests

This chapter includes:

### Concepts

- ▶ [Running Tests Overview](#) on page 184

### Tasks

- ▶ [How to Run a Test](#) on page 187
- ▶ [How to Prepare and Run a Load Test](#) on page 190

### Reference

- ▶ [Command Line Syntax](#) on page 194
- ▶ [Run Test User Interface](#) on page 195
- ▶ [Debug Tabs User Interface](#) on page 197

---

---

## Concepts

---

---

### Running Tests Overview

You can run tests directly from the Service Test interface or from the command line. The test run allows you to determine if the activities are functional and if they are performing as expected.

Before running the test, you can select a location in which to store the results, and set input parameter values specifically for this test run. For details, see "How to Run a Test" on page 187.

For details about the command line, see "Command Line Syntax" on page 194.

The Run Results Viewer provides a detailed report about each step and its checkpoints. For details, see Chapter 7, "Run Results Viewer."

When you run custom code from the user interface, you can also use breakpoints and other tools to debug your code.

Service Test allows you set the run configuration—**Debug** or **Release**. The **Release** mode is more efficient, and is recommended for Load Testing.

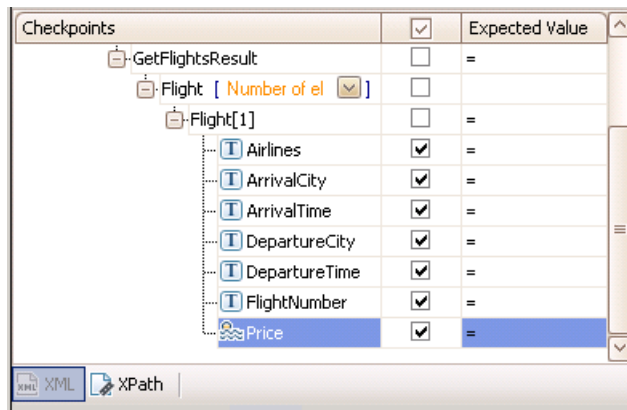
When a test step fails, Service Test continues to run the other test steps. The exception to this is if you enabled the **Stop test if checkpoint fails** option and the checkpoint failed. For details, see "Input/Checkpoints View" on page 70.



## Checkpoint Validation

In functional testing, it is necessary to check the response from the server to confirm that your service performed the actions correctly. The response can contain several properties, each containing several data items. The **Checkpoints** section is a central point for defining the required response values for your test.

Before replaying a test, you set expected values for the test properties. You can enter the values manually, link them to a data source, or load values that were captured during replay. This is useful when you have many argument values—instead of manually entering values, you automatically load them.



Service Test automatically displays all of the step's properties with a check box. To include a checkpoint in the test, you select its check box. If you load replay values, you can select only those properties that you want to check.

The **Xpath** tab lets you validate results against XPATH expressions.

Service Test's Checkpoint setting lets you indicate whether or not to stop the test if the checkpoint fails. For details, see the "Input/Checkpoints View" on page 70.

After the test run, you can view the log or the run results and determine if the results were as expected. The Run Results Viewer displays the checkpoint status in its own sub-node. For details, see "HTTP Checkpoint Response" on page 218.

## Load Testing

You can add load testing capabilities to your test, that will allow you to measure performance under load.

When you have a full LoadRunner installation on the Service Test machine, Service Test provides a custom template, **Service Test Enabled for Load Test**.

For tests created with the standard **Service Test** template, Service Test provides a conversion utility which converts the script into a LoadRunner compatible script, with a .usr extension.

---

**Note:** Tests created in Service Test and converted into LoadRunner scripts, cannot be edited in LoadRunner's Virtual User Generator (VuGen). To edit a test, modify it in Service Test.

---

To measure performance within LoadRunner, you can insert Start and End Transaction markers. During execution, LoadRunner measures the execution time of all the actions between the transaction markers.

When working with data tables, you can use the standard data retrieval methods that are available in LoadRunner. For details, see "Data Retrieval Options for Load Test Enabled Tests" on page 176.

For task details, see "How to Prepare and Run a Load Test" on page 190.

---



---

## Tasks

---



---

### How to Run a Test

This task describes how to run a test through the user interface and view its results.

For information on running a test through the command line, see "Command Line Syntax" on page 194.

This task includes the following steps:

- "Configure checkpoints" on page 187
- "Select a location for the run results - optional" on page 188
- "Save the test in ALM/QC - optional" on page 188
- "Select a location for the run results - optional" on page 188
- "Set values for the test variables- optional" on page 189
- "Run the test" on page 189
- "Debug the test - optional" on page 189
- "Results" on page 189

#### 1 Configure checkpoints

Checkpoint slet you validate the results. For details, see "Checkpoint Validation" on page 185.



- a** Open the **Input/Checkpoints** view, and provide an expected value for each row in one of the following ways:

- Manually
- Link to Source dialog box.
- **Load from replay** button.



- b** Select a comparison operator, such as =, >, or Regex. These may differ depending on the data type.

- c Select the check boxes in the rows of the properties that you want to validate. Clear the check boxes for the properties you are not validating.
  - d To validate against an XPATH expression, click the **Xpath** tab in the Checkpoints section and paste the expression into the **Xpath Checkpoints** column.
  - e If required, select the **Stop test if checkpoint fails** option.
- 

**Tip:** You can customize and override the checkpoint settings using an event handler. For details, see "How to Write Checkpoint Events" on page 268.

---

For user interface details, see the "Input/Checkpoints View" on page 70.

## 2 Select a build configuration - optional

Choose a build configuration. Select **Build > Set Configuration** to select a configuration: **Debug** or **Release**. During development, it is preferable to use the **Debug** option to allow you access to detailed output information. To conserve resources, such as in Load Testing, use the **Release** option.

## 3 Save the test in ALM/QC - optional

To save results in ALM/QC, you need to first save the actual test in ALM/QC. Select **File > HP ALM/QC Connection**.

## 4 Select a location for the run results - optional



- a Click the Run button on the toolbar and select the **Results Location** tab.

To show and hide this dialog box, choose **Edit > Settings > Show Run Dialog Box**.

- b** For test saved on the File System, select a location in which to store the run results—**New run results folder** or **Temporary run results folder**. Choose the latter if you do not expect to use and analyze the results. For details, see "Results Location" on page 196.
- c** For tests saved test in ALM/QC, specify a Run name. If relevant, select a **Test set** and **instance**.

## 5 Set values for the test variables- optional

Select the test properties or user/system variables. For details, see "How to Define Test Properties or User/System Variables" on page 16.

## 6 Run the test

Click the **OK** button in the Run dialog box.

## 7 Debug the test - optional



Use the Break mode and other debug tools to solve any runtime issues.

For user interface details, see the "Service Test Main Window" on page 53 or the "Debug Menu" on page 197.

## 8 Results

View the results in the Run Results Viewer. The viewer opens automatically after a test run.

- A green check mark indicates success.
- A red **X** mark indicates a failure in one of the steps. Expand the nodes of the report to view details about each step and checkpoint.

To view the report at a later stage, choose **Test > View Test Results**.

## How to Prepare and Run a Load Test

This task describes how to prepare a test for load testing in LoadRunner. For an explanation about load testing, see "Load Testing" on page 186.

This task includes the following steps:

- "Prerequisite" on page 190
- "Create a load test" on page 190
- "Add test steps" on page 191
- "Prepare for load testing - optional" on page 191
- "Assign data to the test - optional" on page 192
- "Set the data retrieval properties - optional" on page 192
- "Set the run configuration to Release" on page 193
- "Run the test in Load Test mode" on page 193
- "Results" on page 193

### 1 Prerequisite

Make sure that the HP LoadRunner Controller is installed on the same machine as Service Test. The Load Testing template will be unavailable without a full LoadRunner installation.

### 2 Create a load test

Create a new test with the **Service Test Enabled for Load Test** template.

If you have a test created with the standard Service Test template, select **Test > Enable Test for Load Testing** or click the **Enable Test for Load Testing** button.



### 3 Add test steps

Drag activities from the toolbox into the canvas to add steps to the test.

### 4 Prepare for load testing - optional

To measure the performance of certain steps, define a transaction

- a** Drag the **Start Transaction** activity from the Toolbox's Load Testing category, into the canvas. Place it before the first step you want to measure.
- b** Drag the **End Transaction** activity at the end of the series of steps you want to measure.
- c** Click the Start Transaction step in the canvas, and open the **Input/Checkpoints** view in the Property Sheet. Enter a **Transaction name**. This name will appear in the reports.
- d** Click the End Transaction step in the canvas, and open the **Input/Checkpoints** view in the Property Sheet. Enter the same **Transaction name** as above.
- e** In the End Transaction's Input properties, select a **Status** for reporting: PASS, FAIL, AUTO. or STOP.

---

**Note:** The End Transaction status is only the LoadRunner transaction's status-not the status of step in Service Test. For example, if you assign a **Failed** status to the transaction, Service Test can still issue a **Passed** status for the test step.

---

## 5 Assign data to the test - optional

Import or create data tables for the input properties. For details, see "How to Assign Data to Test Steps" on page 161.

## 6 Set the data retrieval properties - optional

If you have data in the **Data Window** pane, set the data retrieval properties. Click the Link to a Data Source button. For details, see the "Data Retrieval Options for Load Test Enabled Tests" on page 176.

**Select Link Source**

Constant value
  Previous step
  Data query column
  Environment

Data Explorer
 

- Excel
- SampleA
  - Samp

CustomerName	NumberOfTickets
FlightNumber	Class

Show data columns of matching data types only

Attach the query to the loop: Test Flow

Use a unique value for each Virtual User when load testing

Query parameter: Test FlowSampleAppDataInput\_Param  
 Update value on: Each Iteration  
 When out of values: Abort Vuser

Allocate Virtual User values in the controller
 

- Automatically allocate block size
- Allocate 1 values for each Virtual User

Add

Expression:

```
<Body xmlns="http://schemas.xmlsoap.org/soap/envelope/"><CreateF
```

OK Cancel

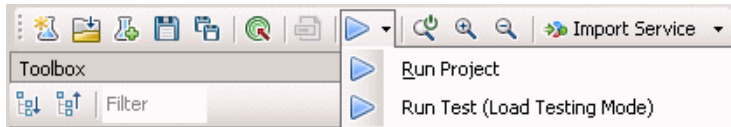


## 7 Set the run configuration to Release

Select **Build > Set Configuration** and select **Release**. The **Release** mode conserves resources, thus enhancing the load testing capabilities.

## 8 Run the test in Load Test mode

Expand the **Run** button and choose **Run test (Load Testing mode)**.



## 9 Results

Check the **Output** tab to verify that the test was successful.

---

**Note:** When you run a test in Load Testing mode, the Run Results Report does not open. To view a Run Results report, choose **Run Project** to run the test in functional mode.

---

---

---

## Reference

---

---

### Command Line Syntax

The Service Test executable, **ServiceTestExecuter.exe**, is located in the product's bin directory.

You can run **ServiceTestExecuter.exe** from the command line using the following parameters:

UI Elements (A-Z)	Description
<b>-test</b>	The full path of the test name (required)
<b>-inParams</b>	The full path of an XML file containing the input parameter values (optional).
<b>-outParams</b>	The full path of an XML file containing the output parameter values (optional).
<b>-profile</b>	The name of an Test profile (optional). For details, see "How to Define Test Properties or User/System Variables" on page 16.
<b>-report</b>	The folder in which to store the report.

For example:

```
C:\Program Files\HP\Service Test\bin> ServiceTestExecuter.exe -test
"c:\MyTests\Test1\Test1" -inParams "c:\MyData\inParams.xml" -outParams
"c:\MyData\outParams.xml"
-profile Profile1
```

where the input parameter file, **InParams.xml** has, for example, this structure:

```
<Arguments>
  <a>1</a>
  <b>2</b>
</Arguments>
```

---

**Tip:** To retrieve a list of all of available parameters, run **ServiceTestExecuter.exe** without any parameters.

---


## Run Test User Interface

This section includes (in alphabetical order):

- ▶ [Run Dialog Box](#) on page 195
- ▶ [Debug Menu](#) on page 197
- ▶ [Debug Output Window](#) on page 199

## Run Dialog Box

Enables you to view the properties of the application.

To Access	 or <b>Debug &gt; Run</b>
Relevant Tasks	"How to Run a Test" on page 187

The following tabs are included:

## Results Location

This dialog box lets you specify a location for the Run Results. The fields differ for a test saved on the file system and one saved in an ALM/QC project.

GUI Element (A-Z)	Description
<b>New run results in HP ALM/QC Project</b>	<p>Stores the run results in the current ALM/QC project. You cannot change the destination <b>Project name</b> (Available when there is a connection to an ALM/QC server).</p> <p>The following fields are editable:</p> <ul style="list-style-type: none"> <li>➤ <b>Run name.</b> A custom name for the test run.</li> <li>➤ <b>Test set.</b> The name of the test set in ALM/QC.</li> <li>➤ <b>Instance.</b> The instance number of the run.</li> </ul>
<b>New run results folder</b>	<p>Stores the run results in a specific folder (Available when there is no connection to an ALM/QC server).</p> <p>Click the Browse button to select a location. The default path is <b>C:\Documents and Settings\%USERPROFILE%\My Documents\ServiceTestProjects\&lt;Test Name Folder\&lt;TestName&gt;</b>.</p>
<b>Temporary run results folder</b>	<p>Stores the run results in a temporary folder, overwriting any existing results. This folder uses the <b>TEMP</b> variable defined for the current user, <b>%USERPROFILE%\Local Settings\Temp\STTempResults</b>.</p>
<b>Don't show this dialog again</b>	<p>Bypasses the Run dialog box in future test runs, using your last selection.</p> <p><b>Tip:</b> To show this dialog box again, select <b>Edit &gt; Settings &gt; Show Run Dialog Box</b>.</p>

## Input Parameters

This tab lets you view and specify values for Input parameters that you defined earlier.

GUI Element (A-Z)	Description
<b>Schema</b>	The name of the parameter.
<b>Value</b>	An editable column containing the parameter values.

For details about defining user variables, see "Action Buttons - Test Variables View" on page 77.

## Debug Tabs User Interface

This section includes (in alphabetical order):







- ▶ [Run Dialog Box](#) on page 195
- ▶ [Debug Menu](#) on page 197
- ▶ [Debug Output Window](#) on page 199


## Debug Menu

Enables you to view the properties of the application.

<b>To Access</b>	<b>Debug</b> menu
<b>Relevant Tasks</b>	"Debug the test - optional" on page 189

The following tabs are included:

GUI Element (A-Z)	Description
	<b>Run.</b> Runs and compile the test.
	<b>Stop Process.</b> Stops the test execution.
	<b>Break.</b> Pauses the test run and enters Break mode in order to allow you to edit the code.
	<b>Continue Debugging.</b> Resumes the test run from the point of the break.
	<p><b>Step Over.</b> Executes the current statement and enters Break mode in order to allow you to edit the code. If the statement calls another procedure, Service Test will not step into that procedure. It will run the procedure, and advance to the next statement in the current procedure.</p> <p><b>Step Into.</b> Executes the current statement and enters Break mode in order to allow you to edit the code. If the statement calls another procedure, Service Test will step into that procedure.</p> <p><b>Step Out.</b> If you used <b>Step Into</b> and the control moved to a called procedure, Step Out automatically runs the current procedure, and returns to the procedure from which it was called.</p>
	<b>Step Into.</b> Executes the current statement and enters Break mode in order to allow you to edit the code. If the statement calls another procedure, Service Test will step into that procedure.






GUI Element (A-Z)	Description
	<b>Step Out.</b> If you used <b>Step Into</b> and the control moved to a called procedure, Step Out automatically runs the current procedure, and returns to the procedure from which it was called.
<b>Toggle Breakpoint</b>	<p>Inserts or removes a breakpoint at the location of the cursor.</p> <p><b>Note:</b> This applies when working in a code window, such as an event, but not in the canvas.</p> <p><b>Tip:</b> The shortcut key for toggling a breakpoint is <b>F7</b>.</p>

## Debug Output Window

Enables you to view the run's debug information.

<b>To Access</b>	<b>View &gt; Debug &gt; ...</b>
<b>Relevant Tasks</b>	"Debug the test - optional" on page 189

The following tabs are included:

GUI Element (A-Z)	Description
<b>Breakpoints tab</b>	<p>A list of the breakpoints in the test, for each open buffer. This tab provides the following controls:</p> <ul style="list-style-type: none"> <li>▶  . <b>Go to Previous Bookmark.</b> Jumps to previous bookmark in the list.</li> <li>▶  . <b>Go to Next.</b> Jumps to next bookmark in the list.</li> <li>▶  . <b>Enable/Disable.</b> Enable or disable all bookmarks.</li> <li>▶  . <b>Delete.</b> Delete the selected bookmark.</li> <li>▶  . <b>Delete All.</b> Delete all bookmarks in all buffers.</li> </ul> <p><b>Tip:</b> To create a bookmark, click on a line within a buffer and press <b>F7</b>.</p>
<b>Callstacks tab</b>	A list of the calls in the stack sorted by <b>Name</b> or <b>Language</b> .

GUI Element (A-Z)	Description
<b>Console tab</b>	A console allowing you to manually enter debug commands
<b>Loaded modules tab</b>	A list of the currently loaded modules by <b>Name, Address, Path, Order, and Symbols.</b>
<b>Local variables tab</b>	A list of the local variables by <b>Name, Value, and Type.</b>
<b>Threads tab</b>	A list of the currently loaded modules by <b>ID, Name, Location, Priority, and Frozen.</b>
<b>Watch tab</b>	<p>A list of watched expressions. The shortcut menu provides the following controls:</p> <ul style="list-style-type: none"> <li>▶ <b>Add watch.</b> Allows you to add an expression to be watched.</li> <li>▶ <b>Remove Watch.</b> Removes the selected watched item.</li> <li>▶ <b>Remove all.</b> Removes all items from the watch list.</li> </ul>



# 6

---

## ALM/QC Integration

This chapter includes:

### Concepts

- ▶ [Managing Tests Through HP ALM/QC Overview](#) on page 202
- ▶ [ALM/QC Tests and Version Control](#) on page 203

### Tasks

- ▶ [How to Work with Tests in ALM/QC](#) on page 204
- ▶ [How to Save Tests to ALM/QC Projects](#) on page 206
- ▶ [How to View a List of Previous Test Versions](#) on page 207

### Reference

- ▶ [ALM/QC User Interface](#) on page 208

---

---

## Concepts

---

---

### **Managing Tests Through HP ALM/QC Overview**

Service Test integrates with supported versions of HP ALM/QC (Application Management Lifecycle/Quality Center). ALM/QC integration allows you maintain projects with a different kinds of tests (such as Service Test and QuickTest tests, business process tests, manual tests, tests created using other HP products, and so forth) that cover all aspects of your application's functionality. Each test in your project is designed to fulfill a specified testing requirement of your application. To meet the goals of a project, you organize the tests in your project into unique groups.

HP ALM/QC provides an intuitive and efficient method for scheduling and running tests, collecting results, analyzing the results, and managing test versions. They also features a system for tracking defects, enabling you to monitor defects closely from initial detection until resolution.

Service Test integrates with Service Test Management, the ALM/QC tool for storing and managing application components, such as Web services. Service Test Management is an extension of ALM/QC and it provides an efficient method for storing and retrieving application components.

In order for Service Test to work with ALM, you must connect to the Web server on which ALM/QC is installed. You can connect to either a local or remote server.

For more information on working with ALM/QC, see the *HP Application Lifecycle Management User Guide*.

## **ALM/QC Tests and Version Control**

If your test is saved in an ALM/QC project that uses version control, the process of opening and saving a test is different.

Tests with version control are either in a state of checked-in or checked-out. When you are working with a test in a checked-out state, any changes you make will not be saved on the ALM/QC server until you check in the test. If you save the test from within Service Test, a temporary file is saved onto your machine that protects the test in case your computer crashes.

If you are working with a test in a checked-in state, the test is read-only and you cannot save your changes until you check out the test.

If a particular test is being saved to ALM/QC for the first time, and the project uses version control, the test automatically starts in a checked-out state.

---

---

## Tasks

---

---

### **How to Work with Tests in ALM/QC**

The following steps describe the workflow of how to work with tests saved in an ALM/QC project.

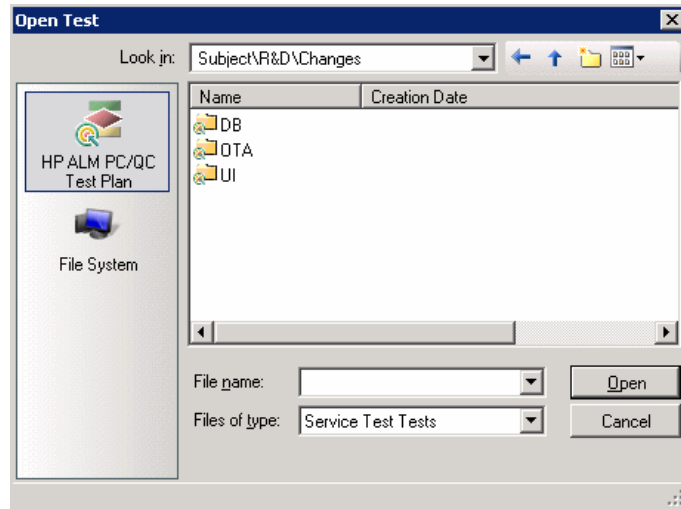
- "Connect to ALM/QC" on page 204
- "Open the test" on page 205
- "(Version control only) Check in/out the test" on page 205
- "(Version control only) Cancel a check out" on page 205
- "Save the test" on page 205

#### **Connect to ALM/QC**

Open a connection to the ALM/QC server. Log into the project that contains the test. For task details, see "How to Connect to ALM/QC" on page 206.

## Open the test

Select **File > New** menu to create a new test, or choose **Open > Test** to open a test already saved in ALM/QC. Select the To find the test, select the **HP ALM/QC Test Lab** module, and drill down in the folder tree until you locate it.



## (Version control only) Check in/out the test

If the ALM/QC project has version control, tests are defined as being either checked-in or checked-out. For more details, see "ALM/QC Tests and Version Control" on page 203. To check in and check out scripts, select **File > ALM/QC Version Control > Check In/Out**.

## (Version control only) Cancel a check out

If you checked out a test and do not want to save the changes, you can return the status of the test to checked-in without saving by selecting **File > ALM/QC Version Control > Undo Check Out**.

## Save the test

Select **File > Save Test As**. Click the **HP ALM/QC Test Plan** button and navigate to the desired location. If you connected to a project that uses version control but is not checked out, the test is only saved as a temporary file on your local machine.

## **How to Connect to ALM/QC**

To store and retrieve scripts from ALM/QC, you need to connect to a project.

- 1** Select **File > HP ALM/QC Connection**.
- 2** Enter a URL, user name and password and click **Connect**. Use the complete URL. For example, <http://servername.dev.lab:8080/qcbin>.

---

**Note:** An [http/https prefix and port number](#) are mandatory.

---

- 3** Select a domain and project for your test and click **Login**.
- 4** To logout from the project, but remain connected to ALM/QC, click **Logout**. This is useful if you want to switch to a different project.
- 5** To disconnect from ALM/QC, click **Disconnect**.

For user interface details, see "HP ALM/QC Connection Dialog Box" on page 208.

## **How to Save Tests to ALM/QC Projects**

The following steps describe how to save a Service Test project to the ALM/QC repository.

- "Connect to ALM/QC" on page 207
- "Open a test" on page 206
- "Save the test to ALM/QC" on page 207

### **Open a test**

Create or open a test. Choose **File > Open > Test** or click the **Create New Test** link on Service Test's **Start Page** tab.

## Connect to ALM/QC

Open a connection to the ALM/QC server and project that you want to store the test. For task details, see "How to Connect to ALM/QC" on page 206.

## Save the test

Choose **File > Save** to save the test to a temporary folder.

## Save the test to ALM/QC



Choose **File > Save As** and select the **HP ALM/QC Test Plan** button. Navigate to the desired location or use the **New Folder** button to create a new directory for the test. Specify a **File Name** and click **Save**.

## How to View a List of Previous Test Versions

If your test is saved in an ALM/QC project that uses version control, you can view a list of the versions of the test. The following steps describe how to do this.

- "Connect to ALM/QC" on page 207
- "Open a test" on page 207
- "View a list of the versions" on page 207

## Connect to ALM/QC

Open a connection to the ALM/QC server and project that you want to store the test. Make sure the ALM/QC project supports versioning.

For task details, see "How to Connect to ALM/QC" on page 206.

## Open a test

Create or open a test. Choose **File > Open > Test**.

## View a list of the versions

Select **File > HP ALM/QC Version Control > Version History** and view the list.

---

---

## Reference

---

---

### **ALM/QC User Interface**

This section includes:

- ▶ [HP ALM/QC Connection Dialog Box](#) on page 208


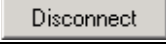
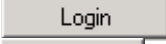
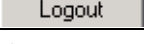
### **HP ALM/QC Connection Dialog Box**

This dialog box enables you to connect to a ALM/QC project from within Service Test.

<b>To access</b>	<b>File &gt; HP ALM/QC Connection ...</b>
<b>Relevant tasks</b>	<ul style="list-style-type: none"><li>▶ "Managing Tests Through HP ALM/QC Overview" on page 202</li><li>▶ "How to Work with Tests in ALM/QC" on page 204</li></ul>



User interface elements are described below:

UI Elements (A-Z)	Description
<b>Step 1: Connect to Server</b>	<ul style="list-style-type: none"> <li>➤ <b>Server URL.</b> The URL of the server that contains ALM/QC.</li> <li>➤ <b>User Name.</b> Your ALM/QC user name.</li> <li>➤ <b>Password.</b> Your ALM/QC password.</li> <li>➤  . Connects to the ALM/QC server with your user information.</li> <li>➤  . Disconnects from the ALM/QC server.</li> </ul>
<b>Step 2: Login to Project</b>	<ul style="list-style-type: none"> <li>➤ <b>Domain.</b> The domain that contains the ALM/QC project. Only those domains containing projects to which you have permission to connect to are displayed.</li> <li>➤ <b>Project.</b> Enter the ALM/QC project name or select a project from the list. Only those projects that you have permission to connect to are displayed.</li> <li>➤  . Logs in to the selected project.</li> <li>➤  . Logs out from the current project.</li> </ul>
<b>Reconnect on startup</b>	<ul style="list-style-type: none"> <li>➤ Connects to ALM/QC the next time you open Service Test.</li> </ul>



# 7

---

## Run Results Viewer

This chapter includes:

### Concepts

- ▶ [Run Results Viewer Overview](#) on page 212
- ▶ [Run Results File Location](#) on page 214
- ▶ [Captured Data](#) on page 215
- ▶ [Custom Fields](#) on page 218

### Tasks

- ▶ [How to Open Run Results](#) on page 220
- ▶ [How to Navigate the Run Results Tree](#) on page 221
- ▶ [How to Manually Submit Defects to Quality Center](#) on page 222
- ▶ [How to Export Run Results](#) on page 224

### Reference

- ▶ [Run Results Viewer User Interface](#) on page 226
- ▶ [Run Results Viewer Commands](#) on page 229
- ▶ [Run Results Viewer Panes](#) on page 232
- ▶ [Run Results Viewer Dialog Boxes](#) on page 245

---

---

## Concepts

---

---

- ▶ [Run Results Viewer Overview](#) on page 212
- ▶ [Run Results File Location](#) on page 214
- ▶ [Captured Data](#) on page 215
- ▶ [Custom Fields](#) on page 218

### **Run Results Viewer Overview**

After running a test, the Run Results Viewer opens automatically at the end of a test run. You can also open it manually at any time.

The Run Results Viewer contains a description of the steps performed during the run session.

- ▶ For a test that does not contain Data Table input parameters, the Run Results Viewer shows a single test iteration.
- ▶ For tests, that contain Data Table input parameters, the Run Results Viewer displays details for each iteration of the test run. The results are grouped by the actions in the test.

For an overview of the various panes, see "Run Results Viewer User Interface" on page 226.

## QuickTest Professional Integration

Service Test lets you perform (UFT) Unified Functional Testing with Quick Test Professional tests. If you run a Service Test test that contains a call to a QuickTest Professional test, or vice versa, you can view the results for all steps performed in the main test and in the called test.

### Run Results XML File

The results of each Service Test run session are saved in a single **.xml** file (called **results.xml**). This **.xml** file stores information on each of the run result nodes in the display. The information in these nodes is used to dynamically create **.htm** files that are shown in the top-right pane in the Run Results Viewer.

Each node in the run results tree is an element in the **results.xml** file. In addition, there are different elements that represent different types of information displayed in the run results. You can take run result information from the **.xml** file and use XSL to display the information you require in a customized format (either when printing from within the Service Test Run Results Viewer, when displaying run results in your own customized results viewer, or when exporting the run results to an HTML file).

XSL provides you with the tools to describe exactly which run result information to display and exactly where and how to display, print or export it. You can also modify the **.css** file referenced by the **.xsl** file, to change the appearance of the report (for example, fonts, colors, and so forth).

For example, in the **results.xml** file, one element tag contains the name of a step, and another element tag contains information on the time at which the run session is performed. Using XSL, you could tell your customized run results viewer that the step name should be displayed in a specific place on the page and in a bold green font, and that the time information should not be displayed at all.

You may find it easier to modify the existing **.xsl** and **.css** files provided with Service Test, instead of creating your own customized files from scratch. The files are located in **C:\Program Files\HP\UFT Result Viewer\dat**, and are named as follows:

- ▶ **PShort.xsl**. Specifies the content of the run results report printed, or exported to an HTML file, when you select the **Short** option in the Print or Export to HTML File dialog boxes.
- ▶ **PDetails.xsl**. Specifies the content of the run results report printed, or exported to an HTML file, when you select the **Detailed** option in the Print or Export to HTML File dialog boxes.
- ▶ **PResults.css**. Specifies the appearance of the run results print preview. This file is referenced by all three **.xsl** files.

For more information on printing run results using a customized **.xsl** file, see "Print Dialog Box" on page 252.

For more information on exporting the run results to a file using a customized **.xsl** file, see "Export Run Results Dialog Box" on page 246.

## Run Results File Location

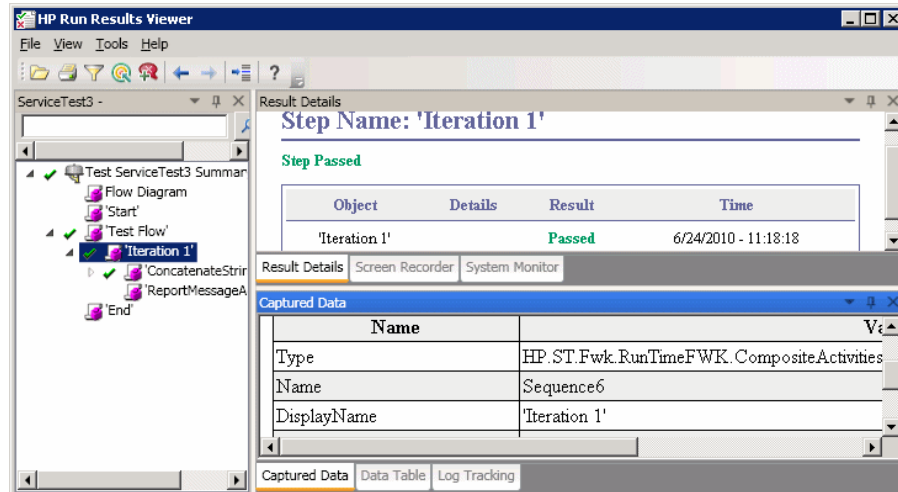
When you run a test, Service Test opens the Run dialog box. You can select a path for the results or use the default location—the test's folder. Alternatively, you can specify a temporary folder, that gets overwritten by each subsequent test run. This is ideal if you are developing a test and have no need for the results.

If your test is stored in ALM/QC (Application Management Lifecycle/Quality Center), the Run dialog box prompts you to store the results in your ALM/QC project. You cannot modify the project, but you can specify a run name, a test folder, and an instance.

For details, see "How to Run a Test" on page 187.

## Captured Data

The captured data tab in the bottom pane shows the data generated during a test run. The contents of the tab differs depending on the level you select in the Results tree in the left pane.



These user interface elements are described below:

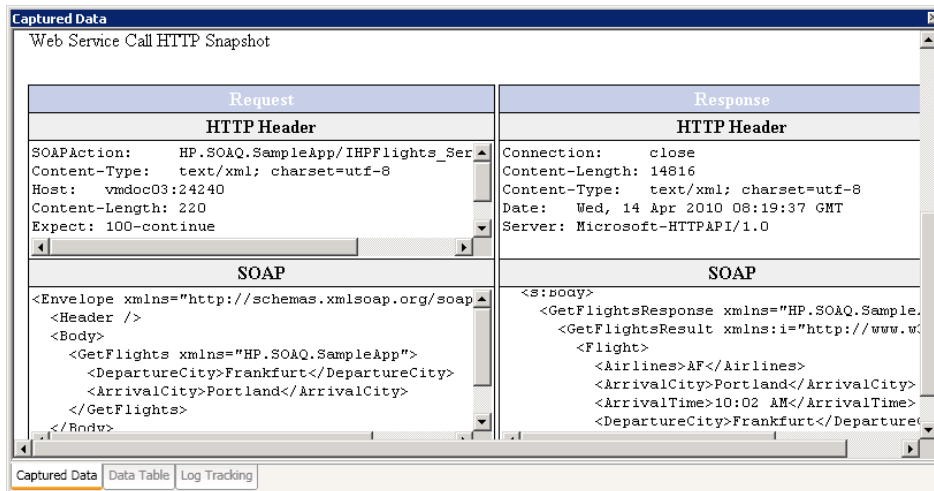
UI Elements	Description
<b>Flow Diagram</b>	A snapshot of the test's canvas.
<b>Start, End</b>	General information about the Start and End activities.
<b>Test Flow / Loop</b>	Information about the loop such as loop name, loop type, display name and so forth.
<b>Iteration</b>	Information about the iteration to which the activities belong.

UI Elements	Description
<b>Activity</b>	<p>Captured data from the activity:</p> <ul style="list-style-type: none"> <li>▶ For Service type activities, this level shows the Request and Response data for the operation or method.</li> <li>▶ For the <b>Report Message</b> activity, this level displays the custom message defined in the activity's properties. For details, see "Sending Messages to the Output and Results Viewer" on page 27.</li> </ul>
<b>Checkpoint</b>	Data about the checkpoints such as the Expected and Actual values, the method of evaluation (Equals, Does Not Equal, and so forth) and status.

## Request and Response

The Activity level captured data contains a table showing the request and response messages.

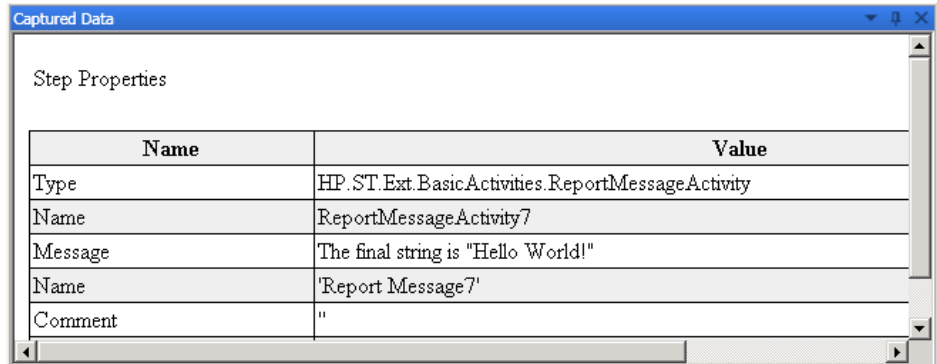
The following example shows the Request and Response data for the **GetFlights** operation from the sample Web Service.





## Messages

For the **Report Message** activity, the Activity level displays a custom message defined in the activity's properties.



The screenshot shows a window titled "Captured Data" with a table labeled "Step Properties". The table has two columns: "Name" and "Value".

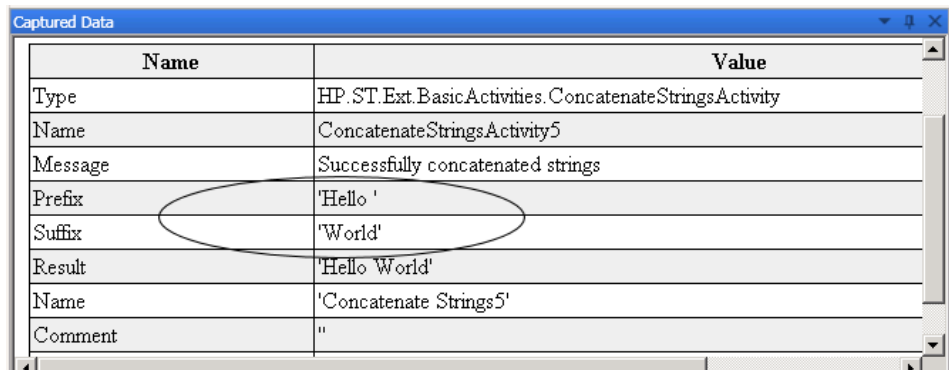
Name	Value
Type	HP.ST.Ext.BasicActivities.ReportMessageActivity
Name	ReportMessageActivity7
Message	The final string is "Hello World!"
Name	'Report Message7'
Comment	"

For details, see "Sending Messages to the Output and Results Viewer" on page 27.

## Data Driven and Parameterized Properties

If you parameterized or applied data driving to an activity's properties, the viewer lists the actual values used during the test run, per iteration.

For built-in activities, the **Step Properties** table shows the values used during the test run.



The screenshot shows a window titled "Captured Data" with a table labeled "Step Properties". The table has two columns: "Name" and "Value". The "Prefix" and "Suffix" rows are circled in the original image.

Name	Value
Type	HP.ST.Ext.BasicActivities.ConcatenateStringsActivity
Name	ConcatenateStringsActivity5
Message	Successfully concatenated strings
Prefix	'Hello '
Suffix	'World'
Result	'Hello World'
Name	'Concatenate Strings5'
Comment	"

For service requests, you can observe the actual values in the **Request/Response** table.

Request	
<b>HTTP Header</b>	
SOAPAction:	HP.SOAQ.SampleApp/IHPFlights_Ser
Content-Type:	text/xml; charset=utf-8
Host:	vmdoc03:24240
Content-Length:	220
Expect:	100-continue
<b>SOAP</b>	
<pre>&lt;Envelope xmlns="http://schemas.xmlsoap.org/soap"   &lt;Header /&gt;   &lt;Body&gt;     &lt;GetFlights xmlns="HP.SOAQ.SampleApp"&gt;       &lt;DepartureCity&gt;Frankfurt&lt;/DepartureCity&gt;       &lt;ArrivalCity&gt;Portland&lt;/ArrivalCity&gt;     &lt;/GetFlights&gt;   &lt;/Body&gt;</pre>	

### HTTP Checkpoint Response

To view the actual result for HTTP activity checkpoints, select the checkpoint's parent node in the **Captured Data** tab, and scroll to the Response body. Locate the actual result string in the Response body table. The child node of the checkpoint does not provide this information.

## Custom Fields

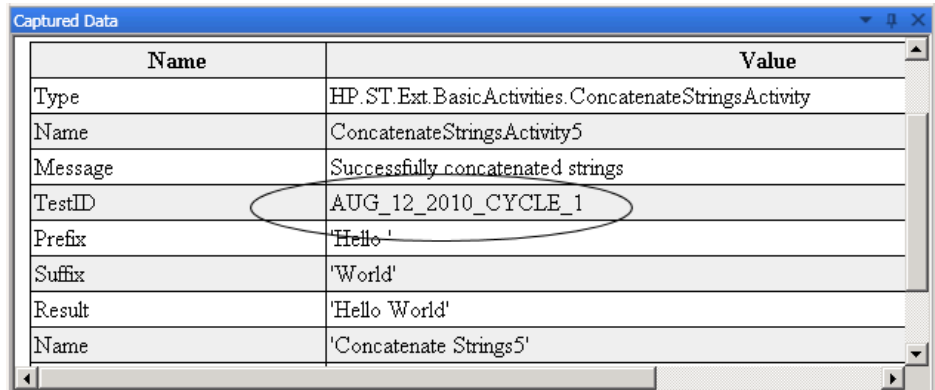
You can use the **Report** function to show custom information in the Results report. You can specify strings or existing arguments and display them in the report.

You add the **Report** function to the step's events. For details, see the section on Custom Code and Events.

The following example prints **AUG-12-2010\_CYCLE\_1** in the Value column for **TestID**, in the Concatenate String step.

```
args.Activity.Report("TestID","AUG-12-2010_CYCLE_1");
```

The report displays the keyword and its value at the Activity level of the results. For details, see the "Captured Data" on page 215.



Name	Value
Type	HP.ST.Ext.BasicActivities.ConcatenateStringsActivity
Name	ConcatenateStringsActivity5
Message	Successfully concatenated strings
TestID	AUG_12_2010_CYCLE_1
Prefix	'Hello '
Suffix	'World'
Result	'Hello World'
Name	'Concatenate Strings5'

---

---

## Tasks

---

---

- ▶ [How to Open Run Results](#) on page 220
- ▶ [How to Navigate the Run Results Tree](#) on page 221
- ▶ [How to Manually Submit Defects to Quality Center](#) on page 222
- ▶ [How to Export Run Results](#) on page 224

### **How to Open Run Results**

After a test run, Service Test automatically opens the Run Results Window with the results of the current run. This task describes how to open specific run results in the Run Results Viewer.

This task includes the following steps:

- ▶ "Open the Run Results Viewer" on page 220
- ▶ "Connect to your Quality Center project - optional" on page 220
- ▶ "View saved results" on page 220

#### **Open the Run Results Viewer**

- ▶ If the Run Results Viewer is not open, select **Test > View Test Results** in the Service Test window.

#### **Connect to your Quality Center project - optional**

If your run results are saved in Quality Center, connect to your Quality Center project before opening the results file. For details, see "Quality Center Connection - Server Connection Dialog Box" on page 255.

#### **View saved results**



Click the **Open** button or select **File > Open**. The Open Run Results dialog box opens. Browse to the relevant results folder or file. For details, see "Open Run Results Dialog Box" on page 251.

## How to Navigate the Run Results Tree

This task describes how to collapse or expand a branch in the run results tree to select the level of detail that the tree displays.

When you open the Run Results Viewer for the first time, the tree expands one level at a time. If the child branches under a parent branch were previously expanded, that state is maintained when you expand or collapse the parent branch.

### To expand a specific branch, do any of the following:

- ▶ Select the branch and click the expand (+) sign to the left of the branch icon.
- ▶ Press the plus key (+) on your keyboard number pad.

The tree displays the details for the branch, and the expand sign changes to collapse.

### To expand a branch and all branches below it:

- ▶ Select the branch and press the asterisk (\*) key on your keyboard number pad.

### To expand all of the branches in the run results tree, do any of the following:

- ▶ Right-click a branch and select **Expand All**.
- ▶ Select the top level of the tree and press the asterisk (\*) key on your keyboard number pad.

### To collapse a specific node, do any of the following:

- ▶ Select it and click the collapse (-) sign to the left of the node icon.
- ▶ Press the minus key (-) on your keyboard number pad.

The node's child nodes disappear from the tree, and the collapse sign changes to expand (+).

### To collapse all of the nodes in the tree:

Right-click a node and select **Collapse All**.

**To move between previously selected nodes within the run results tree:**



Click the **Go to Previous Node** or **Go to Next Node** buttons.

**To find specific steps within the Run Results:**

Use the **Search** box (shown in the example below).



You can search for text and/or types of nodes. For details, see "Run Results Tree Pane and Search Box" on page 232.

**To filter the tree to display only nodes that match certain criteria:**

Use the **Filter dialog box (View > Filters)**. For details, see "Filter Dialog Box" on page 249.

## **How to Manually Submit Defects to Quality Center**

This task describes how to manually add defects to a Quality Center project.

This task includes the following steps:

- "Prerequisites" on page 223
- "Connect to a Quality Center project" on page 223
- "Open the New Defect dialog box" on page 224
- "Modify the defect information if needed and submit it" on page 224
- "Results" on page 224

## 1 Prerequisites

Ensure that the Quality Center client is installed on your computer. (Enter the Quality Center Server URL in a browser and ensure that the Login screen is displayed.)

## 2 Connect to a Quality Center project



Select **Tools > Quality Center Connection** or click the **Quality Center Connection** button and connect to a Quality Center project.

"Quality Center Connection - Server Connection Dialog Box" on page 255.

---

**Note:** If you do not connect to a Quality Center project before proceeding to the next step, Service Test prompts you to connect before continuing.

---

### 3 Open the New Defect dialog box



Select **Tools > Add Defect** or click the **Add Defect** button to open the New Defect dialog box in the specified Quality Center project. The New Defect dialog box opens.

### 4 Modify the defect information if needed and submit it

Basic information is included in the description, but you can modify the defect if needed.

### 5 Results

The defect is added to the Quality Center project's defect database.

## How to Export Run Results

This task describes how to export run results to a file. For details on what is included when you export run results, see "Export Run Results Dialog Box" on page 246.

This task includes the following steps:

- "Open the results in the Run Results Viewer" on page 225
- "Specify the export settings" on page 225
- "Save the file" on page 225
- "Results" on page 225



## 1 Open the results in the Run Results Viewer

For details, see "Open Run Results Dialog Box" on page 251.

## 2 Specify the export settings

Select **File > Export To File**. The Export Run Results dialog box opens. For details on the various settings, see "Export Run Results Dialog Box" on page 246.

## 3 Save the file

Click **Export**. The Save As dialog box opens. Specify the file name and path, and select the required file type.

Report type	Save as type
Step details	<ul style="list-style-type: none"> <li>▶ HTML (*.htm, *.html) (default)</li> <li>▶ PDF (*.pdf)</li> <li>▶ DOC (*.doc)</li> </ul>
Data Table	Excel (.xls)
Log Tracking	XML (.xml)
Screen Recorder	FlashBack (*.fbr)
System Monitor	<ul style="list-style-type: none"> <li>▶ Text (*.csv, *.txt) (default)</li> <li>▶ Excel (.xls)</li> <li>▶ XML (.xml)</li> <li>▶ HTML (*.htm, *.html)</li> </ul>

## 4 Results

When you click **Save**, the file is exported in the specified format to the designated location.

---

---

## Reference

---

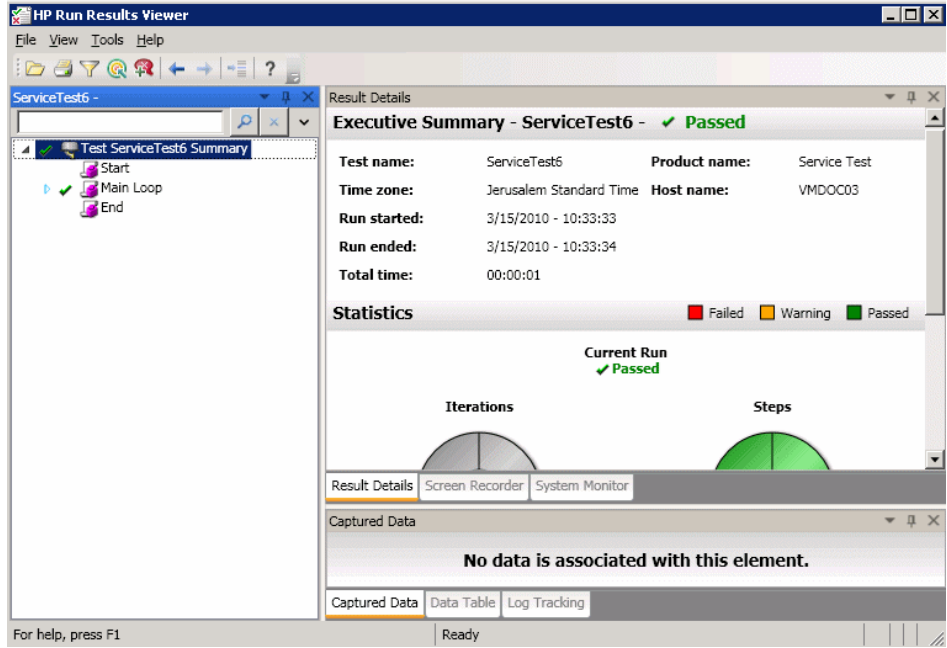
---

- ▶ [Run Results Viewer User Interface](#) on page 226
- ▶ [Run Results Viewer Commands](#) on page 229
- ▶ [Run Results Viewer Panes](#) on page 232
- ▶ [Captured Data Pane](#) on page 240
- ▶ [Run Results Viewer Dialog Boxes](#) on page 245

### **Run Results Viewer User Interface**

This window enables you to view the results of a test run.

The following example shows an Executive Summary:



<p><b>To access</b></p>	<ul style="list-style-type: none"> <li>▶ Opens automatically after a test run. To enable or disable, select <b>Edit &gt; Settings &gt; Show Results After Run</b></li> <li>▶ Choose <b>Test &gt; View Test Results</b></li> </ul>
<p><b>Relevant tasks</b></p>	<p>"How to Run a Test" on page 187</p>

By default, the left pane contains the Run Results tree. The right side of the window contains additional panes. These user interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements	Description
<Run Results Viewer menu bar and toolbar>	See "Run Results Viewer Commands" on page 229.
<b>Executive Summary pane</b>	<ul style="list-style-type: none"> <li>▶ A high-level results overview report (pass/fail status)</li> <li>▶ Quality Center information for your test (if the test was run from Quality Center)</li> </ul>
<b>Run Results Tree pane</b>	<ul style="list-style-type: none"> <li>▶ A graphical representation of the results in an expandable tree</li> <li>▶ Displays the test or component steps, specifying exactly where application failures occurred</li> </ul>
<b>Result Details pane</b>	Detailed explanations of each step and checkpoint pass or failure, at each stage of the test or component
<b>Captured Data pane</b>	A still image of the state of your application at a particular step. For details, see "Captured Data Pane" on page 240.
<status bar>	Displays the status of the currently selected command.



---


**Note:** Certain panes are only available in QuickTest Professional.






---

## Run Results Viewer Commands

The Run Results Viewer menu bar and toolbar contain commands to help you view run session results.

Button	Command	Shortcut Key	Description
	<b>File &gt; Open</b>	CTRL+O	Opens the Open Run Results dialog box, enabling you to open saved run results from the file system or from Quality Center. For details, see "How to Open Run Results" on page 220.
	<b>File &gt; Print</b>	CTRL+P	Opens the Print dialog box, enabling you to print the results of the run session. For details, see "Print Dialog Box" on page 252.
--	<b>File &gt; Print Preview</b>	CTRL+F2	Opens the Print Preview dialog box, enabling you to preview the results of the run session prior to printing. For details, see "Print Preview Dialog Box (Run Results Viewer)" on page 254.
--	<b>File &gt; Export To File</b>	--	Opens the Export Run Results dialog box, enabling you to save various parts of the results as external files. For details, see "Export Run Results Dialog Box" on page 246.
--	<b>File &gt; Recent Files</b>	--	Lists the recently viewed files.
--	<b>File &gt; Exit</b>	--	Closes the Run Results Viewer session.
--	<b>View &gt; Run Results Viewer Toolbar</b>	--	Shows or hides the Run Results Viewer toolbar.

Button	Command	Shortcut Key	Description
--	<b>View &gt; Status Bar</b>	--	Shows or hides the status bar, which indicates: <ul style="list-style-type: none"> <li>▶ a hint about the currently selected command</li> <li>▶ the status of the Run Results Viewer</li> <li>▶ the Quality Center server name and project to which the Run Results Viewer is connected</li> </ul>
--	<b>View &gt; Result Details</b>	--	Shows or hides the Results Details pane. For details, see "Result Details Pane" on page 236.
--	<b>View &gt; Captured Data</b>	--	Shows or hides the Captured Data pane. For details, see "Captured Data Pane" on page 240.
--	<b>View &gt; Restore Layout</b>	--	Restores the default layout of the Run Results Viewer.
	<b>View &gt; Filters</b>	CTRL+T	Opens the Filters dialog box, enabling you to filter the information displayed. For details, see the "Filter Dialog Box" on page 249.
--	<b>View &gt; Expand All</b>	--	Expands all of the branches in the run results tree. Also available as a context-menu option.
--	<b>View &gt; Collapse All</b>	--	Collapses all of the branches in the run results tree. Also available from the shortcut menu.

Button	Command	Shortcut Key	Description
	<b>Tools &gt; Add Defect</b>	--	Enables you to add a defect to your Quality Center project. If you are not currently connected to Quality Center, opens the Quality Center Connection - Server Connection dialog box. For more information, see "How to Manually Submit Defects to Quality Center" on page 222.
	<b>Tools &gt; Quality Center Connection</b>	--	Opens the Quality Center Connection - Server Connection dialog box, enabling you to connect to a Quality Center project. For more information, see the "Quality Center Connection - Server Connection Dialog Box" on page 255.
	<b>Go to Previous Node</b>	BACKSPACE	Moves the cursor to the previously selected node in the run results tree. For more information, see "How to Navigate the Run Results Tree" on page 221.
	<b>Go to Next Node</b>	ALT+RIGHT ARROW	Moves the cursor to the node you selected in the run results tree prior to clicking the <b>Go to Previous Node</b> button. For more information, see "How to Navigate the Run Results Tree" on page 221.
	<b>Help &gt; Help Topics</b>	--	Opens the online help for the Run Results Viewer.
--	<b>Help &gt; About Run Results Viewer</b>	--	Displays version information about the Run Results Viewer.

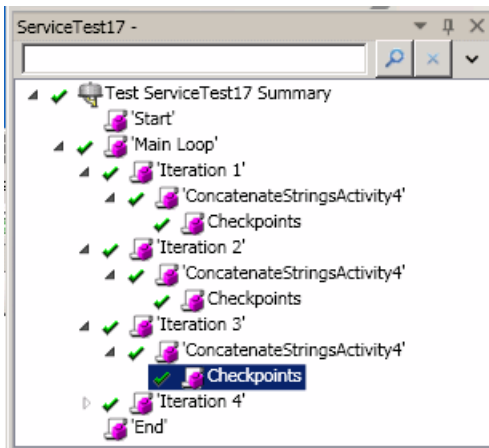
## Run Results Viewer Panes

This section includes:

- ▶ "Run Results Tree Pane and Search Box" on page 232
- ▶ "Result Details Pane" on page 236

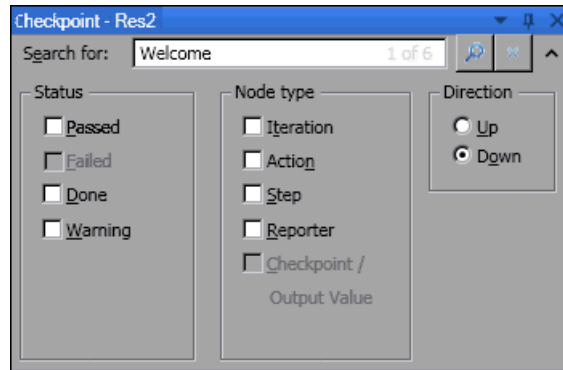
### Run Results Tree Pane and Search Box

This pane displays the **run results tree**—a hierarchical representation of the test run results.







The following image shows an example of an expanded **Search** box. Six instances of the searched for text, **Welcome**, were found.



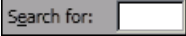




<b>To access</b>	This pane is displayed by default on the left side of the Run Results Viewer. It cannot be hidden.
<b>Important information</b>	You can collapse or expand a node in the run results tree to change the level of detail that the tree displays. You can also use the Search and Filter commands to control what is displayed in the run results tree. For details, see "Run Results Tree Pane and Search Box" on page 232 and "Filter Dialog Box" on page 249.

User interface elements are described below:

UI Elements	Description
✓	A step that succeeded. <b>Note:</b> If a test does not contain checkpoints, no icon is displayed.
✗	A step that failed. Note that this causes all parent steps (up to the root action, test, or component) to fail as well.
!	A warning, meaning that the step did not succeed, but it did not cause the action, test or component to fail.

UI Elements	Description
	A step that failed unexpectedly.
	The run session was stopped before it ended.

**Search** box user interface elements are described below:

UI Elements	Description
	Text box in which you can optionally enter text for which to search.  If the specified text is found in one or more tree nodes, the text area indicates this, as shown below:    In this example, 1 of 6 indicates that there are six nodes displaying the text, Welcome, and the first matching node is highlighted in the tree.
	<b>Search.</b> Finds the next instance that matches the criteria you specified in the <b>Search</b> box. Click this button to jump to each node that matches your search criteria.
	<b>Cancel.</b> Clears the <b>Search for</b> text box.
	<b>Expand</b> or <b>Collapse.</b> Shows or hides the lower part of the <b>Search</b> box.

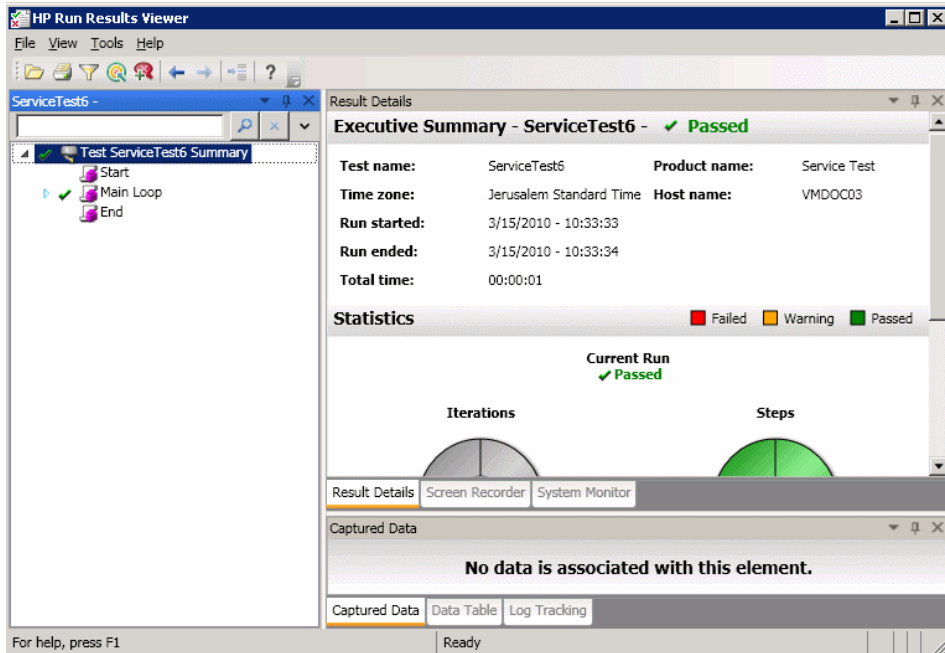
UI Elements	Description
<b>Status</b>	<p>The status to search for. (Optional)</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>▶ <b>Passed.</b> Searches for steps that passed and match your other selection criteria.</li> <li>▶ <b>Failed.</b> Searches for steps that failed and match your other selection criteria.</li> <li>▶ <b>Done.</b> Searches for steps with the status <b>Done</b> (steps that were performed successfully but did not receive a pass, fail, or warning status) that match your other selection criteria.</li> <li>▶ <b>Warning.</b> Searches for steps with the status <b>Warning</b> (steps that did not pass, but did not cause the test to fail) that match your other selection criteria.</li> </ul> <p><b>Note:</b> If the tree does not contain any steps that match a particular status, that option is grayed out in the <b>Search</b> box.</p>
<b>Node type</b>	<p>The type of node to search for (together with your other search criteria). (Optional)</p> <p>Possible values for Service Test:</p> <ul style="list-style-type: none"> <li>▶ <b>Step.</b> Searches for steps that match your other selection criteria.</li> </ul>
<b>Direction</b>	<p>The direction to search in the tree. Possible values:</p> <ul style="list-style-type: none"> <li>▶ <b>Up</b></li> <li>▶ <b>Down</b></li> </ul>

## Result Details Pane

This pane displays the details for an individual iteration, an action, or a step that is currently selected in the run results tree.

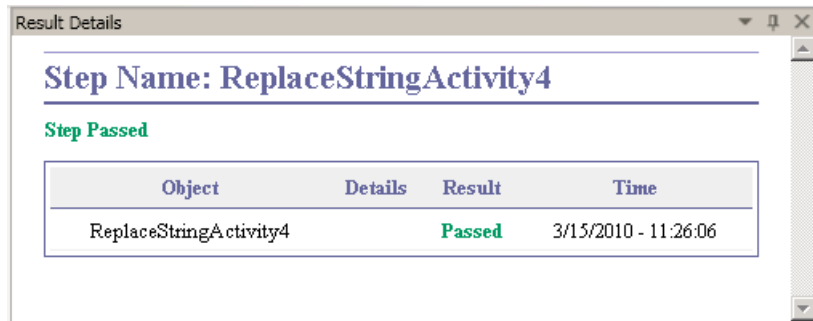
### Example of Executive Summary:

The Executive Summary is displayed when the topmost node in the run results tree is selected.



**Example of Result Details:**

Result details are displayed when any node (other than the topmost node) is selected in the run results tree.



<b>To access</b>	<p>Do the following:</p> <ol style="list-style-type: none"> <li><b>1</b> Select a node in the run results tree: <ul style="list-style-type: none"> <li>➤ To open the <b>Executive Summary</b> page, select the topmost node in the tree.</li> <li>➤ To open the <b>Result Details</b> for a step, select the relevant node in the tree.</li> </ul> </li> <li><b>2</b> Select the <b>Result Details</b> tab. (This assumes that the default layout is displayed.)</li> </ol> <p><b>Tip:</b> If the Result Details pane is hidden, select <b>View &gt; Result Details</b> to show it.</p>
<b>Important information</b>	<p>By default, when the Run Results Viewer opens after a run session, an <b>Executive Summary</b> is displayed in the Result Details pane.</p> <p>For any other node, the details in the Result Details pane are specific for the step selected in the run results tree. For details, see "Captured Data" on page 215.</p>

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

### Executive Summary Page

UI Elements	Description
<b>Executive Summary</b>	<p>Includes:</p> <ul style="list-style-type: none"> <li>▶ the test or component name and result details</li> <li>▶ time-related information for the run</li> <li>▶ the product from which the test or component was run</li> <li>▶ Quality Center server and project, if Service Test was connected to a Quality Center project during the run</li> </ul> <p><b>Note:</b> If a test or component that is stored in Quality Center is run from QuickTest, but you choose to store the results in a temporary location, the <b>Test set</b> and <b>Test instance</b> fields are not displayed in the results.</p>
<b>Statistics</b>	Graphical, status-related statistics for the current run and the previous run (if any). If the test or component was run previously, you can click <b>Open</b> to open a previous run results.

### Result Details for Step

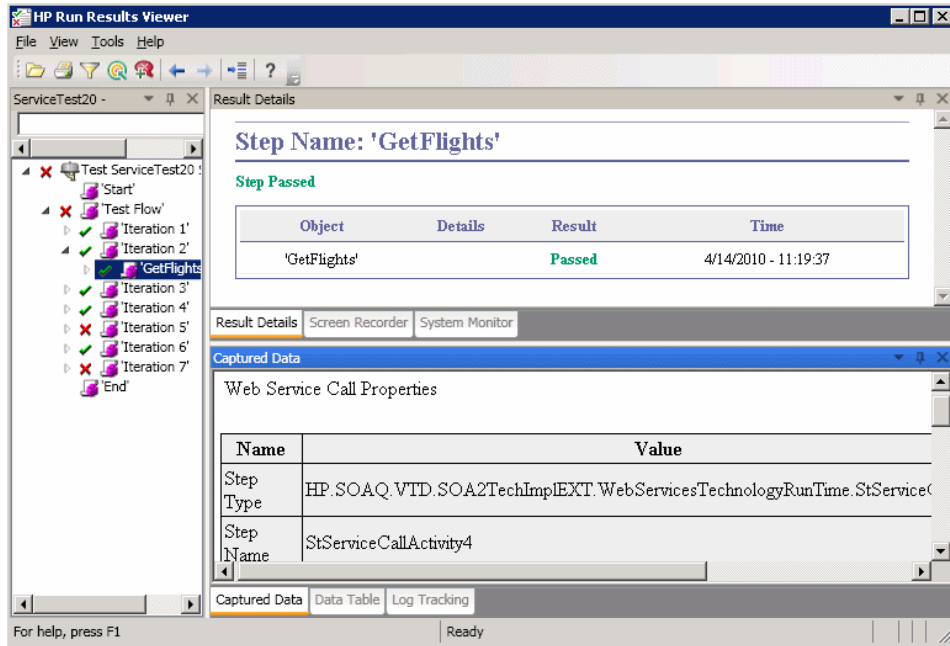
UI Elements	Description
<step name>	The name of the step.

UI Elements	Description
<step status>	<p>The status of the step. Possible values:</p> <ul style="list-style-type: none"> <li>▶ <b>Done.</b> Relevant for iterations, actions, and steps that ran successfully, but do not contain checkpoints.</li> <li>▶ <b>Failed.</b> Relevant for iterations, actions, and steps that contain checkpoints.</li> <li>▶ <b>Passed.</b> Relevant for iterations, actions, and steps that contain checkpoints.</li> <li>▶ <b>Warning.</b> Relevant for steps that were not successful, but did not cause the test or component to stop running.</li> </ul> <p><b>Note:</b> A test, component, iteration, or action containing a step marked <b>Warning</b> may still be labeled <b>Passed</b> or <b>Done</b>.</p>
<step details>	<p>Details about the step, such as the object on which the step was performed, the timestamp, the step results, and so on. The information in this area changes according to the type of step.</p> <p>For more details, see "Captured Data" on page 215.</p>

## Captured Data Pane

The captured data tab in the bottom pane shows the data generated during a test run. The contents of the tab differs depending on the level you select in the step pane tree in the left pane.

The following image shows the Captured Data pane showing the Web Service Call properties.



<b>To access</b>	Select <b>View &gt; Captured Data</b> or click the <b>Captured Data</b> tab.
<b>Important information</b>	<b>Programmatically adding information to the results:</b> You can programmatically add items to the report. For details, see "Custom Fields" on page 218.
<b>Relevant tasks</b>	"How to Navigate the Run Results Tree" on page 221



The contents of the Captured Data in Service Test pane differs depending on the level you select in the run results tree.

- ▶ **Start, End.** General information about the Start and End activities.
- ▶ **Parent step.** Information about the loop to which the test steps belong, such as **Test Flow**.
- ▶ **Iteration.** Information about the iteration to which the activities belongs such as the status of the iteration and its display name.
- ▶ **Activity.** Captured data from the activity, or operation in the case of Web Services. For Service type activities, this level shows the Request and Response data.
- ▶ **Checkpoints.** Data about the checkpoints such as the Expected and Actual values, the method of evaluation (Equals, Does Not Equal, and so forth) and status.

## Samples of Captured Data

### Request and Response

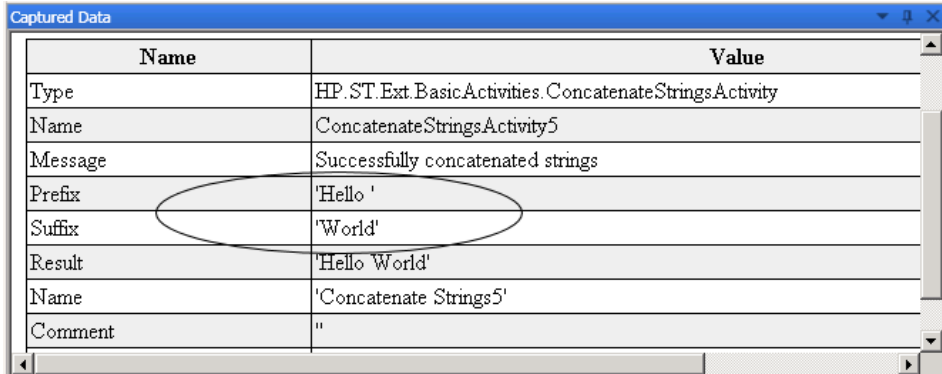
The following example shows the Request and Response data for the **GetFlights** operation from the sample Web Service, on the Step level.

Request	Response
<p><b>HTTP Header</b></p> <p>SOAPAction: HP.SOAQ.SampleApp/IHPFlights_Ser            Content-Type: text/xml; charset=utf-8            Host: vmdoc03:24240            Content-Length: 220            Expect: 100-continue</p>	<p><b>HTTP Header</b></p> <p>Connection: close            Content-Length: 14816            Content-Type: text/xml; charset=utf-8            Date: Wed, 14 Apr 2010 08:19:37 GMT            Server: Microsoft-HTTPAPI/1.0</p>
<p><b>SOAP</b></p> <pre>&lt;Envelope xmlns="http://schemas.xmlsoap.org/soap"   &lt;Header /&gt;   &lt;Body&gt;     &lt;GetFlights xmlns="HP.SOAQ.SampleApp"&gt;       &lt;DepartureCity&gt;Frankfurt&lt;/DepartureCity&gt;       &lt;ArrivalCity&gt;Portland&lt;/ArrivalCity&gt;     &lt;/GetFlights&gt;   &lt;/Body&gt;</pre>	<p><b>SOAP</b></p> <pre>&lt;s:body&gt;   &lt;GetFlightsResponse xmlns="HP.SOAQ.Sample.   &lt;GetFlightsResult xmlns:i="http://www.w   &lt;Flight&gt;     &lt;Airlines&gt;AF&lt;/Airlines&gt;     &lt;ArrivalCity&gt;Portland&lt;/ArrivalCity&gt;     &lt;ArrivalTime&gt;10:02 AM&lt;/ArrivalTime&gt;     &lt;DepartureCity&gt;Frankfurt&lt;/DepartureC</pre>

## Data Driven and Parameterized Properties

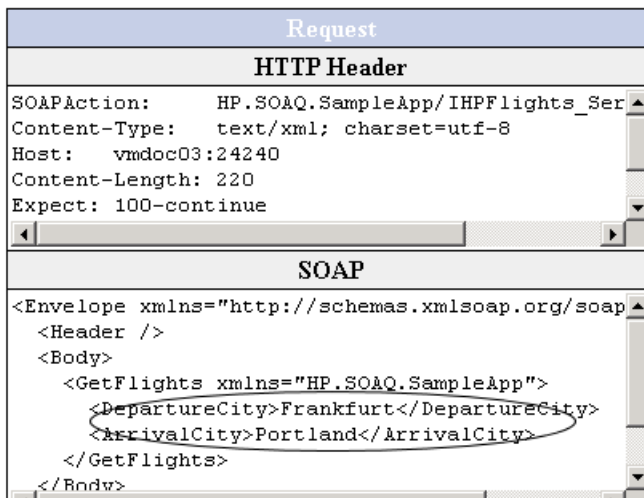
If you parameterized or applied data driving to an activity's properties, the viewer lists the actual values used during the test run, per iteration.

For built-in activities, the **Step Properties** table shows the values used during the test run.



Name	Value
Type	HP.ST.Ext.BasicActivities.ConcatenateStringsActivity
Name	ConcatenateStringsActivity5
Message	Successfully concatenated strings
Prefix	'Hello '
Suffix	'World'
Result	'Hello World'
Name	'Concatenate Strings5'
Comment	"

For service requests, you can observe the actual values in the **Request/Response** table.



Request	
<b>HTTP Header</b>	
SOAPAction:	HP.SOAQ.SampleApp/IHPFlights_Ser
Content-Type:	text/xml; charset=utf-8
Host:	vmdoc03:24240
Content-Length:	220
Expect:	100-continue
<b>SOAP</b>	
<pre>&lt;Envelope xmlns="http://schemas.xmlsoap.org/soap"   &lt;Header /&gt;   &lt;Body&gt;     &lt;GetFlights xmlns="HP.SOAQ.SampleApp"&gt;       &lt;DepartureCity&gt;Frankfurt&lt;/DepartureCity&gt;       &lt;ArrivalCity&gt;Portland&lt;/ArrivalCity&gt;     &lt;/GetFlights&gt;   &lt;/Body&gt;</pre>	

## Web Service Activity

The following tables shows the results of a Web Service operation from the Sample application, **GetFlights**:

In the results tree:	In the Result Details pane:
Step Type	The type of step as it is defined in the source code. This field is not editable by the user. For example <code>HP.ST.Ext.WebServicesTechnologyRT.StServiceCallActivity</code> .
Step Name	The step name as it appears in the test's source code, visible when you create an event handler for a step. For example <code>StServiceCallActivity1</code> . This field is read-only.
Service Name	The service name as defined in the WSDL.
Port Name	The name of the operation's port.
Operation Name	The name of the operation.
Endpoint Address	The endpoint for the service to which the requests are sent.
SOAP Action	The full path of the SOAP action, such as <code>HP.SOAQ.SampleApp/IHPFlights_Service/CreateFlightOrder</code> .
Content Type	The content type of the request message, for example, <code>text/xml, charset=utf</code>
Message Exchange Pattern	The pattern of the messages, for example <code>Request {followed by} Response</code> .
HTTP Status Code	The HTTP Status code returned by the server.

## Concatenate String Activity

The following tables shows the results of a the String Manipulation Activity, Concatenate String.

In the results tree:	In the Result Details pane:
Type	The internal type name of the step. <code>HP.ST.Ext.BasicActivities.ConcatenateStringsActivity</code>

In the results tree:	In the Result Details pane:
Name	The step name as it appears in the test's source code, <code>ConcatenateStringsActivity</code> . This is visible when you create an event handler for a step.
Message	A message indicating the status and purpose of the step. For example, "Successfully concatenated strings."
Prefix	The built-in property, <b>Prefix</b> .
Suffix	The built-in property, <b>Suffix</b> .
Result	The resulting text after the step was executed.
Name	The name of the step as it appears in the canvas, <code>ConcatenateStringsActivity</code> , or a custom name.
Comment	The comment entered in the step's <b>General</b> tab.
Status	The run status, such as <b>Done</b> , <b>Failed</b> , and so forth.

### Checkpoint Captured Data

The Checkpoint level captured data is only presented if you enable Checkpoints for the test run. A grid provides the following information for each checkpoint:

In the results tree:	In the Result Details pane:
<b>Name</b>	The checkpoint index. For example, <code>Checkpoint()</code> , <code>Checkpoint1</code> , and so forth.
<b>Result</b>	Passed or Failed, indicated graphically with a check or X mark.
<b>Property</b>	The path of the checkpoint in the SOAP envelope. For example, <code>Envelope[1]\Body[1]\CreateFlightOrderResponse[1]\CreateFlightOrderResult[1]\OrderNumber[1]</code> .
<b>Actual Result</b>	The actual output value received in the response.

In the results tree:	In the Result Details pane:
<b>Evaluation Style</b>	The type of comparison operator, such as <b>EqualTo</b> , and so forth.
<b>Expected Values</b>	The response you expected to receive, from the Checkpoint grid's <b>Expected Value</b> field.
<b>Details</b>	Checkpoint details, when relevant. For example, if you add an <b>Assert</b> function to the step's event handler, the text will appear in this field. For details, see the "How to Write Checkpoint Events" on page 268.

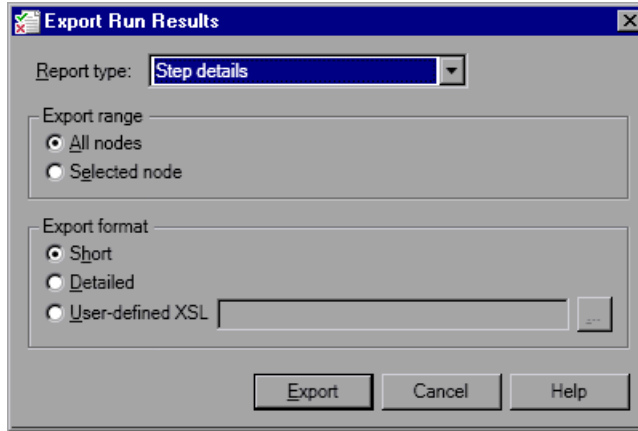
## Run Results Viewer Dialog Boxes

This section includes (in alphabetical order):

- [Export Run Results Dialog Box](#) on page 246
- [Filter Dialog Box](#) on page 249
- [Open Run Results Dialog Box](#) on page 251
- [Print Dialog Box](#) on page 252
- [Print Preview Dialog Box \(Run Results Viewer\)](#) on page 254
- [Quality Center Connection - Server Connection Dialog Box](#) on page 255

## **Export Run Results Dialog Box**

This dialog box enables you to export run results to a file so that you can view them even if the Run Results Viewer is unavailable. For example, you can send the file containing the run results in an e-mail to a third-party who does not have the Run Results Viewer installed.



<b>To access</b>	Select <b>File &gt; Export to File</b> .
<b>Important information</b>	<ul style="list-style-type: none"> <li>▶ <b>The export time varies according to the size of the results file and the file type you select.</b> When selecting the file type, consider the length of time it will take to generate different document types, especially for a report with many images. HTML files generate the fastest, followed by PDF and DOC. When exporting a report with 100 or more images to a DOC file, a dialog box is displayed reminding you that it may take a long time to generate the file. The dialog box gives you the option to continue exporting with images, continue exporting without images, or to export to PDF.</li> <li>▶ <b>Screen capture images are not exported for steps on Web-based applications.</b> When you export run results containing steps on a Web application, any screen capture images for those steps are not exported to the file. This is because for Web-based applications, the Run Results Viewer displays the HTML corresponding to the relevant Web page (with downloaded images) rather than a captured image and thus no image is saved with the report.</li> </ul>
<b>Relevant tasks</b>	"How to Export Run Results" on page 224

User interface elements are described below:

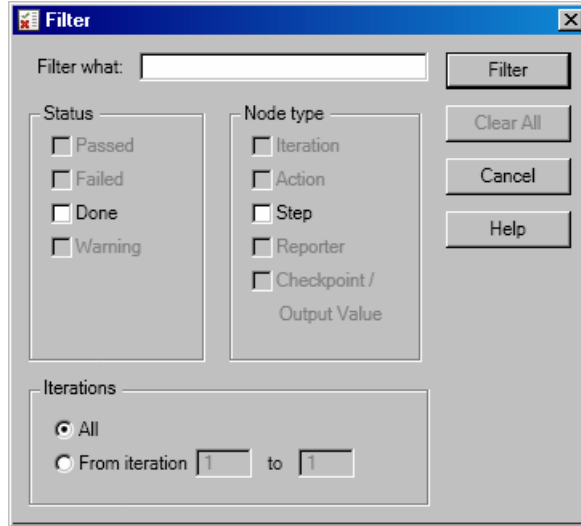
<b>UI Elements</b>	<b>Description</b>
<b>Report type</b>	The type of report you want to export, for example, Step Details.

UI Elements	Description
<b>Export range</b>	<p>Relevant only for <b>Step Details</b> report type.</p> <ul style="list-style-type: none"> <li>▶ <b>All nodes.</b> Exports the results for the entire test or component.</li> <li>▶ <b>Selected node.</b> Exports run result information for the selected branch in the run results tree.</li> </ul>
<b>Export format</b>	<p>Relevant only for <b>Step Details</b> report type.</p> <ul style="list-style-type: none"> <li>▶ <b>Short.</b> Exports a summary line (when available) for each item in the run results tree. This option is only available if you selected <b>All nodes</b> in <b>Export range</b>.</li> <li>▶ <b>Detailed.</b> Exports all available information for each item in the run results tree, or for the selected branch, according to your <b>Export range</b> selection. (In the Run Results Viewer, these images are displayed in the Captured Data pane.)</li> <li>▶ <b>User-defined XSL.</b> Enables you to browse to and select a customized <b>.xsl</b> file. You can create a customized <b>.xsl</b> file that specifies the information to be included in the exported report, and the way it should appear. For more information, see the "Run Results XML File" on page 213.</li> </ul>



## Filter Dialog Box

This dialog box enables you to filter the results tree to display only those nodes that match the conditions that you specify.



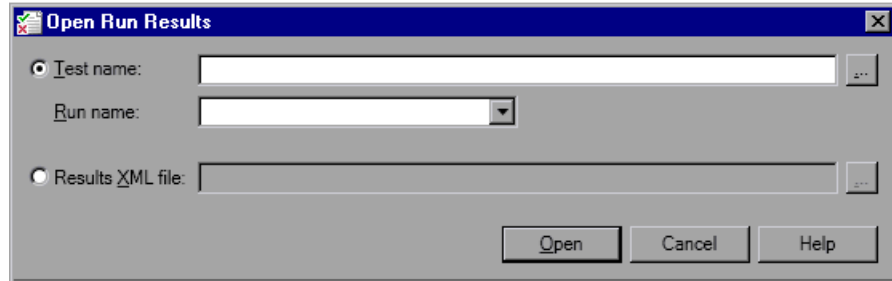
To access	Select <b>View</b> > <b>Filters</b>
-----------	-------------------------------------


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

UI Elements	Description
<Filter box>	The text by which you want to filter.
<b>Status</b>	<ul style="list-style-type: none"> <li>➤ <b>Passed.</b> Displays the run results for the steps that passed.</li> <li>➤ <b>Failed.</b> Displays the run results for the steps that failed.</li> <li>➤ <b>Done.</b> Displays the run results for the steps with the status <b>Done</b> (steps that were performed successfully but did not receive a pass, fail, or warning status).</li> <li>➤ <b>Warning.</b> Displays the run results for the steps with the status <b>Warning</b> (steps that did not pass, but did not cause the test to fail).</li> </ul>
<b>Node type list</b>	<p>Displays all results that match your selection criteria based on: (Optional)</p> <ul style="list-style-type: none"> <li>➤ <b>Iteration.</b> Displays the run results in the run results tree for the iterations specified in the <b>Iterations</b> area.</li> <li>➤ <b>Action.</b> Displays the run results for all actions in the run results tree that match your other selection criteria.</li> <li>➤ <b>Component.</b> Displays the run results for all components in the run results tree that match your selection criteria.</li> <li>➤ <b>Step.</b> Displays the run results for all steps in the run results tree that match your other selection criteria.</li> <li>➤ <b>Reporter.</b> Displays the run results for all Reporter steps in the run results tree that match your other selection criteria. (QuickTest Professional only)</li> <li>➤ <b>Checkpoint/Output Value.</b> Displays the run results for all checkpoint and output value steps in the run results tree that match your other selection criteria.</li> </ul>
<b>Iterations</b>	<ul style="list-style-type: none"> <li>➤ <b>All.</b> Displays run results from all iterations.</li> <li>➤ <b>From iteration X to Y.</b> Displays the run results from the specified range of test iterations.</li> </ul>

## Open Run Results Dialog Box

This dialog box enables you to select the results to display in the Run Results Viewer.



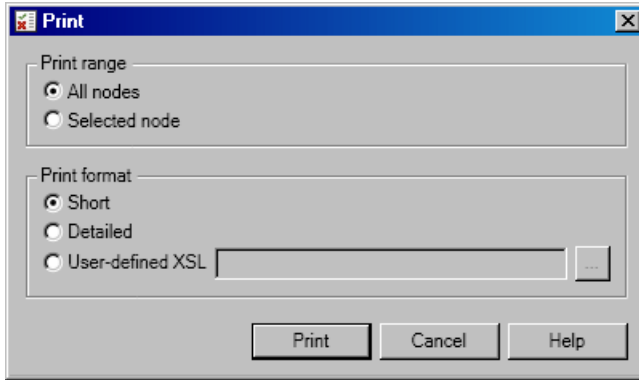
<b>To access</b>	Select <b>File &gt; Open</b> or click the <b>Open</b> button  .
<b>Important information</b>	<ul style="list-style-type: none"> <li>▶ To view results stored in Quality Center, you must first connect to your Quality Center project. For details, see "Quality Center Connection - Server Connection Dialog Box" on page 255.</li> <li>▶ By default, results files for tests are stored in the Reports folder under the user's My Documents folder.</li> </ul>
<b>Relevant tasks</b>	"How to Open Run Results" on page 220


User interface elements are described below.

UI Elements	Description
<b>Test name</b>	The name of the test for which you want to view results. The test can be located in the file system or in a Quality Center project. If you select the <b>Test name</b> option, you must also specify the <b>Results folder name</b> . The Browse button allows you to navigate to the location of the test.
<b>Results folder</b>	The results for a particular run.
<b>Results XML file</b>	An XML file in which the results are stored. The file must be located in the file system.

## **Print Dialog Box**

This dialog box enables you to print run results from the Run Results Viewer. You can select the type of report you want to print, and you can also create and print a customized report.



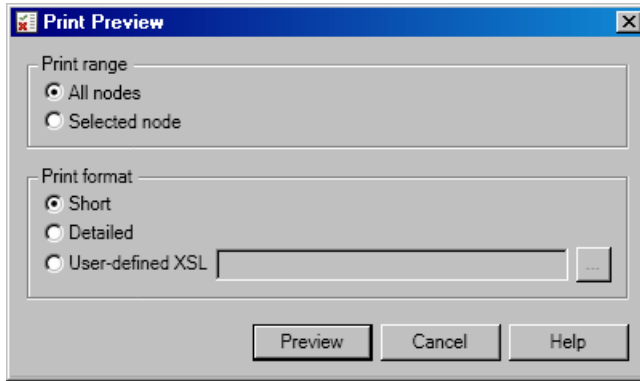
<b>To access</b>	Use one of the following: <ul style="list-style-type: none"><li>▶ Select <b>File &gt; Print</b></li><li>▶ Click the <b>Print</b> button </li></ul>
------------------	---

User interface elements are described below:

UI Elements	Description
<b>Print range</b>	<ul style="list-style-type: none"> <li>▶ <b>All nodes.</b> Prints the run results for the entire test or component.</li> <li>▶ <b>Selected nodes.</b> Prints the run results information for the selected branch in the run results tree.</li> </ul>
<b>Print format</b>	<ul style="list-style-type: none"> <li>▶ <b>Short.</b> Prints a summary line (when available) for each item in the run results tree. This option is only available if you selected <b>All</b> in <b>Print range</b>.</li> <li>▶ <b>Detailed.</b> Prints all available information for each item in the run results tree, or for the selected branch, according to your selection in <b>Print range</b>. The printed report includes still images associated with the steps in your run results. If a bitmap checkpoint step displays expected, actual, and difference bitmaps, these are also included.</li> <li>▶ <b>User-defined XSL.</b> Enables you to browse to and select a customized <b>.xsl</b> file. You can create a customized <b>.xsl</b> file that specifies the information to be included in the printed report, and the way it should appear.</li> </ul>



## Print Preview Dialog Box (Run Results Viewer)

This dialog box enables you to preview run results on screen before you print them. You can select the type and quantity of information you want to view, and you can also display the information in a customized format.



<b>To access</b>	Select <b>File &gt; Print Preview</b>
------------------	---------------------------------------

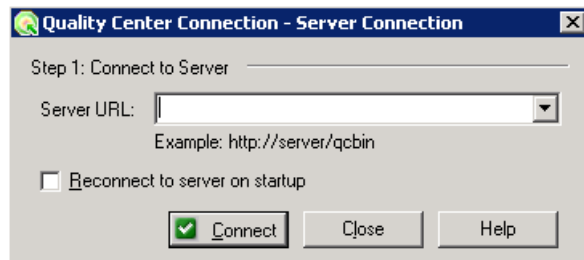
User interface elements are described below:

UI Elements	Description
	<p>Displays the run results on screen as they will appear when printed.</p> <p><b>Tip:</b> If some of the information is cut off in the preview, for example, if checkpoint names are too long to fit in the display, click the <b>Page Setup</b> button  in the Print Preview window and change the page orientation from <b>Portrait</b> to <b>Landscape</b>.</p>

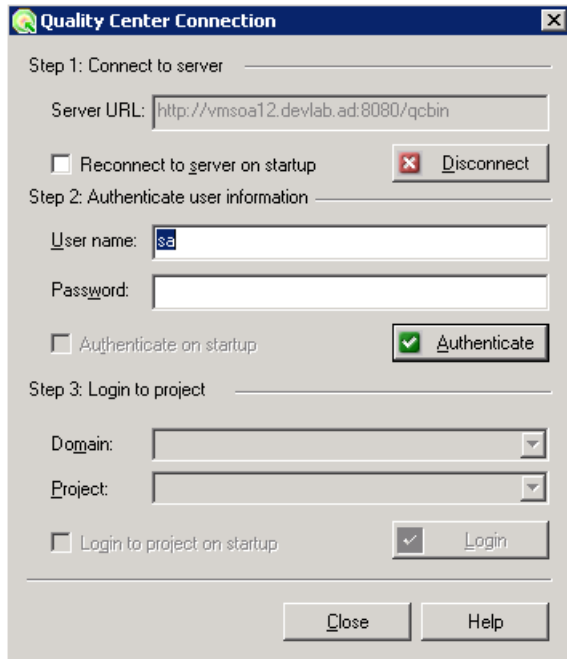
UI Elements	Description
<b>Print range</b>	<ul style="list-style-type: none"> <li>▶ <b>All nodes.</b> Previews the run results for the entire test.</li> <li>▶ <b>Selected nodes.</b> Previews run results information for the selected branch in the run results tree.</li> </ul>
<b>Print format</b>	<ul style="list-style-type: none"> <li>▶ <b>Short.</b> Previews a summary line (when available) for each item in the run results tree. This option is only available if you selected <b>All</b> in <b>Print range</b>.</li> <li>▶ <b>Detailed.</b> Previews all available information for each item in the run results tree, or for the selected branch, according to your selection in <b>Print range</b>. The preview includes still images associated with the steps in your run results. If a bitmap checkpoint step displays expected, actual, and difference bitmaps, these are also included.</li> <li>▶ <b>User-defined XSL.</b> Enables you to browse to and select a customized <b>.xsl</b> file. You can create a customized <b>.xsl</b> file that specifies the information to be included in the preview, and the way it should appear. For more information, see "Run Results XML File" on page 213.</li> </ul>

## **Quality Center Connection - Server Connection Dialog Box**

This dialog box enables you to connect or disconnect to or from a Quality Center project.



After you connect to the server, the dialog box expands to include the remaining connection fields.



<b>To access</b>	Select <b>Tools &gt; Quality Center Connection</b>
<b>Important information</b>	<p><b>Connect.</b> The connection process has two stages. First, you connect to a local or remote Quality Center server. This server handles the connections between Service Test and the Quality Center project.</p> <p>Next, you log in and choose the project you want the Run Results to access.</p>
<b>Relevant tasks</b>	"How to Open Run Results" on page 220
<b>See also</b>	"Run Results Viewer Overview" on page 212



User interface elements are described below:

UI Elements (A-Z)	Description
<b>Authenticate / Change User</b>	<p>Authenticates your user information against the Quality Center server.</p> <p><b>Note:</b> After your user information has been authenticated, the edit boxes in the <b>Authenticate</b> user information area are displayed in read-only format. The <b>Authenticate</b> button changes to a <b>Change User</b> button.</p> <p><b>Tip:</b> You can log in to the same Quality Center server using a different user name by clicking <b>Change User</b>, and then entering a new user name and password and clicking <b>Authenticate</b> again.</p>
<b>Authenticate on Startup</b>	Automatically authenticates your user information against the Quality Center server every time you open the Run Results Viewer.
<b>Close</b>	Closes the Quality Center Connection dialog box.
<b>Connect / Disconnect</b>	Connects or disconnects the Run Results Viewer to or from the selected Quality Center server. (After you successfully connect to a Quality Center server, the button changes to <b>Disconnect</b> .)
<b>Domain</b>	<p>The domain that contains the Quality Center project.</p> <p><b>Note:</b> Only those domains to which you have permission to connect are displayed.</p>
<b>Login / Logout</b>	<ul style="list-style-type: none"> <li>▶ <b>Login.</b> Logs into the selected domain and project using the current user information. (After you successfully log into a project, the button changes to <b>Logout</b>.)</li> <li>▶ <b>Logout.</b> Logs out of the selected domain and project.</li> </ul>
<b>Login to project on startup</b>	Automatically logs into the selected project every time the Run Results Viewer opens.
<b>Password</b>	Your Quality Center password.
<b>Project</b>	<p>The Quality Center project with which you want to work.</p> <p><b>Note:</b> Only those projects for which you are a defined user are displayed.</p>

UI Elements (A-Z)	Description
<b>Reconnect to Server on startup</b>	Automatically reconnects to the Quality Center server every time you open the Run Results Viewer.
<b>Server URL</b>	The URL address of the Web server where Quality Center is installed.  You can choose a Quality Center server accessible via a Local Area Network (LAN) or a Wide Area Network (WAN).
<b>User name</b>	Your Quality Center user name.

### Guidelines for Windows Vista, Windows 7, and Server 2008 Users

The security settings in the following operating systems may prevent you from connecting to a Quality Center project:

- Windows Vista
- Windows 7
- Windows Server 2008

This can occur when the UAC (User Account Control) option is set to ON, and you have never connected to a Quality Center project.

To connect to Quality Center for the first time, you must disable the UAC option. After you successfully connect to Quality Center, you can turn the UAC option on again. Thereafter, you should be able to connect to Quality Center, as needed.

#### To enable Service Test to connect to a Quality Center project:

- 1** Log in as an administrator.
- 2** From the Control Panel, select **User Accounts > Change Security Settings**.
- 3** Clear the **Use User Account Control (UAC) to help protect your computer** check box and click **OK**.
- 4** Connect to Quality Center as described below.
- 5** Select the **Use User Account Control (UAC) to help protect your computer** check box and click **OK** to turn the UAC option on again.





# A

---

## Coding Service Test Events

This chapter includes:

### Concepts

- ▶ [Writing Code for Events Overview](#) on page 262

### Tasks

- ▶ [How to Open a Window for Writing Custom Code](#) on page 264
- ▶ [How to Access Input and Output Properties](#) on page 265
- ▶ [How to Use the Logging Function](#) on page 266
- ▶ [How to Use the Report Function](#) on page 267
- ▶ [How to Write Checkpoint Events](#) on page 268
- ▶ [How to retrieve and set test variables](#) on page 270
- ▶ [How to encrypt and decrypt passwords](#) on page 270
- ▶ [How to use the OnReceiveResponse Web Service Event](#) on page 272

### Reference

- ▶ [Context methods](#) on page 274
- ▶ [Logging Options](#) on page 275
- ▶ [Assert Options](#) on page 276

---

---

## Concepts

---

---

### Writing Code for Events Overview

Service Test lets you write customized code for your test in two ways:

- ▶ Event Handlers for all activities
- ▶ Custom Code activity

For each activity, you can define event handlers for common events, such as: **CodeCheckPointEvent**, **BeforeExecuteStepEvent**, and **AfterExecuteStepEvent**. Certain activity types such as Web Services and HTTP have additional built-in events. For a list of the events, see the "Events View" on page 80.

The **Custom Code** activity allows you to define the behavior of your own activity and its properties. You define this behavior through an event handler that is triggered before, during, or after the execution of the activity.

Service Test's code editor allows you to use autocompletion to prepare the event handler. For task details, see "How to Open a Window for Writing Custom Code" on page 264.

---

**Tip:** To instruct Service Test to ignore an event handler, select the handler and press **DELETE** to remove the name of the handler from the grid.

---

### Casting

The Event Handler code in the **SharedUserCode.cs** file uses the following objects:

- ▶ **sender**. The object that raises the activity's event.
- ▶ **args**. The event arguments passed to the activity.

To access an activity's context, such as any input and output arguments, cast the **args.Activity** object to the appropriate activity type. The built-in template provides commented code which illustrates the casting of the activity to allow you to use its context.

---

---

## Tasks

---

---

### How to Open a Window for Writing Custom Code

For non-custom code activities, you can define default event handlers for checkpoints, before step execution, and after step execution.

The checkpoint event handlers help you verify the output values in your test. You can use a Report, Assert, or Log function to gather information about your service.

#### 1 Open the Events view



Select an activity title in the canvas and open the **Events** view in the Property Sheet.

#### 2 Select an event

Select the event for which you want to provide code. Double-click in the **Handler** column.

#### 3 Edit the code

Click the **SharedUserCode.cs** tab. Locate the **TODO** section and add your custom code. Follow the sample code in the comments and use autocompletion to write your own code.

#### 4 Enable access to the activity's properties

If you need to access an activity's input or output parameters, remove the comment tags from the activity object. For example:

```
// Uncomment the code below to access the activity's properties.  
//var activity = ((HP.ST.Ext.HTTPActivity.HTTPActivity)(args.Activity));  
// TODO: Add your code here...
```



## 5 Save the changes.

Click **File > Save All** to save all buffers of the test.

## How to Access Input and Output Properties

You can set the values of input and output properties through event handlers.

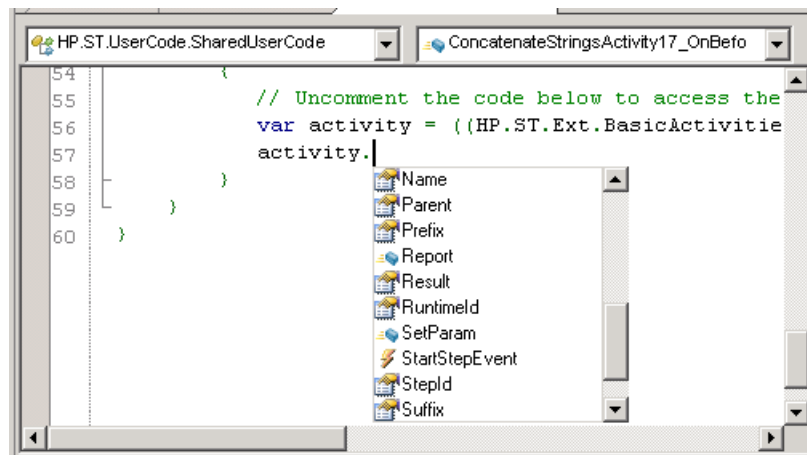
### 1 Access the activity's properties

Access the activity's properties by removing the comments in the code. For example:

```
// Uncomment the code below to access the activity's properties.
var activity = ((HP.ST.Ext.BasicActivities.ConcatenateStringsActivity)(args.Activity));
```

### 2 Select a property

Use autocompletion to locate and select a property from the dropdown list. The dropdown shows all the activity-specific properties. For example, for the **ConcatenateString** activity, the dropdown shows **Prefix** and **Suffix**.



### 3 Access the activity's parent

To access the activity's parent, select the **Parent** method from the dropdown list. A parent refers to the step, condition, or loop that encloses the activity. For example, for a step in a conditional branch, the parent is the name of the condition step.

```
string ParentName = activity.Parent.Name;
```

## How to Use the Logging Function

The **UserLogger** function generates messages during replay. You can view these messages in the **Output** tab or in the log file.

### 1 Select the UserLogger object

Use autocompletion from the activity's **Context** object, and select the **UserLogger** function.

```
args.activity.Context.UserLogger
```

### 2 Choose a log type

Use autocomplete to choose a log type. For a list of the log types, see "Logging Options" on page 275.

### 3 Complete the function

Complete the function as indicated by the tooltip. Provide the necessary arguments.

The following example logs unformatted **Debug** information to the log from a variable you defined earlier called *debug\_information*.

```
activity.Context.UserLogger.Debug(debug_information);
```

## 4 View the Output Log

View the output information in the **Output** tab or in the log file.

**To view the output tab:**

- a** Run the test at least once (F5).
- b** Select **View > Output**. View the Output tab in the bottom section of the Service Test window.

**To view the output log file:**

- a** Run the test at least once (F5).
- b** Select **Test > Open Script Directory** and open the **Log** directory.
- c** Open the `vtd_user.log` file.

## How to Use the Report Function

Use the **Report** function to show the results of a test's execution in the Run Results Viewer. The report opens automatically after you run the test. You can also view it at any time by selecting **Test > View Test Results**.

### 1 Select the Report object

Use the `args.activity` object, and select the **Report** object using autocomplete.

```
args.activity.Report
```

### 2 Complete the function

Provide a keyword and value, based on the function's prototype:  
`Report(string keyword, string value);`

The following example sends the employee's last name to the report using a variable that was defined earlier called *LastName*.

```
args.Activity.Report("Employee Name",LastName);
```

### 3 View the run results

Run the test and view the Run Results. The report displays the keyword and its value in the iteration's branch.

## How to Write Checkpoint Events

You write checkpoint event handlers using the **checkpoint** object. The Assert library functions run a comparison between actual and expected values, and show the results in the step's **Checkpoint** node. You can also use the **checkpoint** object to control checkpoints in the user interface and customize your reports.

### 1 Create a checkpoint handler

Double-click the **Handler** column of the **CodeCheckPointEvent** row in the activity's **Events** view.

### 2 Access the activity's properties

Access the activity's properties by removing the comments in the code. For example:

```
// Uncomment the code below to access the activity's properties.
var activity = ((HP.ST.Ext.BasicActivities.ConcatenateStringsActivity)(args.Activity));
```

### 3 Select the checkpoint object

Use the **args** object, and select the **Checkpoint** object using autocomplete.

```
args.Checkpoint
```

### 4 Validate a checkpoint - optional

To validate a value during run-time, follow these steps:

- a** Use the **args.checkpoint** object, and select the **Assert** object using autocomplete.
- b** Use autocomplete to select a comparison operator such as Equals, and so forth. For details, see "Assert Options" on page 276.

- c Complete the function as indicated by its tooltip. Provide the necessary arguments.

The first example checks if the actual value (first argument) equals the expected value (second argument) for the **ConcatenateString** activity.

```
args.Checkpoint.Assert.Equals(activity.Prefix+activity.Suffix, activity.Result);
```

The second example checks if the text value (alphabetical order for a string) of the prefix is less than the suffix. To ensure that the checkpoint succeeds, enter a prefix value that is greater than the suffix, for example a prefix of **aa** and a suffix of **bb**.

```
args.Checkpoint.Assert.Less("Concatenate test", activity.Prefix, activity.Suffix,
    "Prefix is less than suffix");
```

## 5 Ignore selected checkpoints - optional

You can instruct Service Test to ignore the checkpoints selected in the Service Test UI (Input/Checkpoints view). This is useful if you need to ignore the checkpoints temporarily.

- a Use the **args.checkpoint** object, and select the **RunUICheckpoints** object using autocomplete.
- b Set the value to **false**. You can enable the checkpoints for this step at any time by removing this line or by setting it to **true**.

```
args.Checkpoint.RunUICheckpoints = false;
```

## 6 Customize the checkpoints reporting - optional

You can instruct Service Test to send a specific message to the Checkpoints node in the report. The standard Report function sends a message to the step's node, but the Checkpoint's Report function allows you to send messages to the Checkpoint sub-node.

- a Use the **args.checkpoint** object, and select the **Report** object using autocomplete.
- b Complete the function syntax, based on the function's tooltip, providing the actual and expected result and a logical operator.

## 7 View the results

Run the test and view the Run Results Viewer. Drill down to the checkpoints and check your results.

## How to retrieve and set test variables

The **Context** object's **TestProfile** method allows you to set or get the value of a user test variable from the active Test Profile.

### 1 Select the TestProfile object

Use the **args.activity.Context** object, and select **TestProfile** from the autocomplete menu.

### 2 Get or Set a Variable

From the autocomplete menu, select **GetVariableValue** or **SetVariableValue** methods.

The following example retrieves the value of the *TestName* test variable.

```
string testName = activity.Context.TestProfile.GetVariableValue("TestName");
```

## How to encrypt and decrypt passwords

Password fields expect an encrypted string—if you provide an unencrypted string, the authentication will fail.

The **EncryptionMngr** method lets you encrypt and decrypt strings within your events.

## 1 Encrypt the password

Use the `args.activity.Context.EncryptionMngr` method, select **Encrypt** from the autocompletion dropdown. The following example encrypts a password.

```
// Original password
string plainText = "myPassword";
args.Activity.Context.EncryptionMngr.Encrypt("plainText");
...
```

## 2 Decrypt the password - optional

Use the **Decrypt** method from the autocompletion dropdown.

The following example decrypts a password and validates against the original string.

```
args.Activity.Context.EncryptionMngr.Decrypt("encryptedText");
...
// Check that the decrypted string is the same as the original password
Debug.Assert(plaintext.Equals(decryptedText));
```

## How to use the `OnReceiveResponse` Web Service Event

Service Test provides several custom event handlers for Web Services. The following example shows how to send the inner SOAP envelope as a string to the execution report, when a response is received.

---

**Note:** To manipulate the Output Envelope (`activity.OutputEnvelope` property), use the `AfterExecuteStepEvent`—the `activity.OutputEnvelope` property is not available for the `OnReceiveResponse` event.

---

For a complete list of the Web Service events, open the Property Sheet's **Events** view for a Web Service's step.

### 1 Create an event handler

Select a Web Service step and double-click the **Handler** column of the `OnReceiveResponse` row in the activity's **Events** view. A code editor containing the new event handler will open.

### 2 Access the activity's properties

Access the activity's properties by removing the comments in the code.

```
// Uncomment the code below to access the activity's properties.
var activity =
  ((HP.ST.Ext.WebServicesActivities.StServiceCallActivity)(args.Activity));
```

### 3 Define an `XmlDocument`

Use autocompletion to select the `XmlDocument` method. Define a name for the envelope and assign it the value of `activity.OutputEnvelope`.

```
XmlDocument xdoc = activity.OutputEnvelope
```



#### 4 Send the report to the XmlDocument

Add a call to **args.Activity.Report**, pass in a short description, and the inner text of the XML document that represents the envelope.

```
args.Activity.Report("Envelope Inner Text",xdoc.InnerText);
```

---



---

## Reference




---



---

### Context methods

This section describes some of the common **Context** methods.

UI Elements	Description
	Denotes a function.
	Denotes a string.
	Denotes an event.
<b>EncryptionMngr</b>	Allows you to encrypt and decrypt password strings.
<b>TestInputParameters</b>	Allows you to retrieve the test's input parameters.
<b>TestIteration</b>	Provides information about the test iteration.
<b>TestOutputParameters</b>	Allows you to retrieve the test's output parameters.
<b>TestProfile</b>	Allows you to set and retrieve user variables. For details on how to define user variables, see "How to Define Test Properties or User/System Variables" on page 16.

For a complete list of the functions, use the autocompletion feature:

## Logging Options

The Service Test API provides the following logging options, available from the **Context.UserLogger** object:

UI Elements	Description
<b>Debug, DebugFormat</b>	Logs debug information collected during the test run.
<b>Error, ErrorFormat</b>	Logs errors that occurred during the test run.
<b>Fatal, FatalFormat</b>	Logs fatal errors that occurred during the test run.
<b>Info, InfoFormat</b>	Logs informational data from the test run.
<b>Warn, WarnFormat</b>	Logs warnings that were issued during the test run.

The **Format** option allows you to provide values based on the list of items. The following example logs formatted informational messages, using the **FirstName** variable's value for the first item, and the **LastName** variable for the second item, and so forth.

```
activity.Context.UserLogger.InfoFormat("Hello: {0}{1}", FirstName, LastName);
```

For task details, see "How to Use the Logging Function" on page 266.

## **Assert Options**

The Assert object provides the following comparison operators:

UI Elements
<b>Equals</b>
<b>Greater</b>
<b>GreaterOrEqual</b>
<b>Less</b>
<b>LessOrEqual</b>
<b>NotEquals</b>

For task details, see "How to Write Checkpoint Events" on page 268.

---

# Index

## A

- absolute path 15
- activities
  - adding to the test 42
  - in toolbox 29
- Advanced Settings dialog box (security) 137
- advanced settings, security scenario 109
- alerts 26
- algorithm, for encryption 111
- ALM/QC
  - connecting to 206
  - integration 201
  - managing Vuser scripts 202
  - version control 203
- API 261
- Assert function 268
- attach query 173, 178
- attachments
  - adding 44
  - view in Property Sheet 81
- authentication
  - certificate 108
  - WSDL import 134

## B

- binary encoding 109
- Bootstrap policy 106
- build configurations 184

## C

- Call Java Class Settings dialog box 95
- Captured Data pane 240
- casting 262
- certificate authentication 108
- checkpoints

- configuring 187
- HTTP 218
- keywords 157
- overview 185
- child relations
  - defining 155
  - dialog box 181
- class, Java types 33
- coding events 261
- collapse all 59
- Context object
  - TestProfile 270
  - UserLogger 266
- control flow 25
- copy and paste 25
- Create New Test dialog box 19
- Custom Binding scenario 84, 88
- custom binding scenario 103
- custom fields 218
- custom properties 42

## D

- data
  - assign 161
  - child relations 155
  - create queries 158
  - defining 148
  - from data driving 151
  - from data source 150
  - keywords 156
  - navigation 153, 164
  - populating manually 149
- data driving
  - data 151
  - settings 167
- data navigation 178

- Data Navigation dialog box 164
- Data Queries view 82
- data query
  - Excel 168
  - local table 169
- data relation 181
- data sources 29, 150
- Debug configuration 184
- Debug menu 197
- debugging window 199
- decryption 270
- defects, reporting
  - from Run Results 222
- digital signatures 102
- do while loop 72
- documentation, online 11

## E

- Edit Elements dialog box 170
- encoding, for WS config file 109
- encrypted data for Web Services security 102
- encryption 270
- end transaction 186
- endpoint address 33
- entropy mode 111
- error messages 94
- Errors tab 94
- event handler
  - assert function 268
  - checkpoints 45
  - coding overview 262
  - ignoring 262
  - Report function 267
  - writing 261
- Excel file
  - add query 168
  - as a data source 29
- EXISTS keyword 156
- expand all 59
- Export Run Results dialog box 246
- extensibility 29

## F

- Federation security scenario
  - details 90
  - with STS 140
- flow control 25
- for loop 72
- foreign key 155

## H

- handler event
  - OnReceiveResponse 272
- HTTP Binding 91
- HTTP checkpoints 218

## I

- identities, Web Service security 104
- Import Service from URL/UDDI dialog box 134
- importing services
  - file system 133
  - from URL 134
- Installation Guide 11
- IsOneWay 33

## J

- Java settings 95
- Java testing 32
- Java types, support for 33
- Java virtual machine 66
- JMS test settings 66
- JNDI settings 66
- JVM settings 66

## K

- keywords 156

## L

- load testing
  - preparing test 190
  - transactions 186
- LoadRunner integration 36
- local data table 29

local table, add query 169  
 loop settings 72

## M

message signatures 102  
 messages 94  
 messages, sending to report 217  
 MTOM 109  
 mutual certificates 142, 143

## N

namedPipe 140  
 navigating data 178  
 navigating in data 153  
 navigation properties 164  
 negative testing, Web Services 114  
 netTcp 140  
 nettcp 143  
 NIL 156  
 NOT\_FOUND 156

## O

online documentation 11  
 online help 11  
 Output tabs 93, 199

## P

Print dialog box, Run Results Viewer 252  
 Print Preview dialog box (Run Results Viewer) 254  
 profiles 17, 18  
 properties  
   additional 42  
   selecting data source 174  
 Protection Level 111

## Q

Quality Center 206  
   reporting defects 222  
 query  
   attaching 173, 178  
   child 181

data 168  
 local data table 169

QuickTest Professional integration  
   adding steps 47  
   results 213  
   testing 36

## R

Readme 11  
 realm, Web Service security 113  
 relative path 15  
 Release configuration 184  
 Reliable Messaging 109  
 replay, preparing for 98  
 Report function 267  
 Report Message activity 216  
 REST service testing 31  
 results See run results  
 Run dialog box 195  
 run results  
   custom fields 218  
   display 213  
   exporting to a file 246  
   filtering 222  
   finding 222  
   previewing before printing 254  
   printing 252  
   reporting defects manually 222  
   Run Results window 226  
   tree 232  
 Run Results window, toolbar 229  
 running tests 183

## S

save, save as command 54  
 scenario setting, advanced 109  
 security, Web Service scenario 100  
 security elements, order 120  
 security scenarios  
   Custom Binding 84, 88  
   Federation 90  
   how to define 139  
   HTTP Binding 91  
   user interface 84

- security settings, advanced 137
- security, examples for Web Services 129
- Select Certificate dialog box 138
- Select Link Source dialog box 174
- Select Service dialog box 133
- Select Service from UDDI 135
- send request to service 72
- Service Test
  - introducing 23
  - main window 53
  - working with 21
- services
  - importing 134
  - importing from file system 133
  - references 98
  - testing 23
- signatures, Web Service messages 102
- SKIP 156
- SOAP fault view 76
- SOAP faults 114
- SOAP request, manual 98
- software updates 10
- SPN 105
- SSL, testing Web Service 144
- start transaction 186

## T

- TCP transport 143
- test results, see run results
- Test Settings view 66
- test variables 270
  - defining 17, 18
- Test Variables view 77
- tests
  - creating 13
  - overview 14
  - See also run results
- Tests pane 57
- Toolbox 29
- transactions 186
- transport
  - customizing HTTP 113
  - JMS 36

## U

- UDDI, importing services from 135
- UFT testing 36
  - how to implement 47
  - results 213
- updates 10
- UPN 105
- user variables 17, 18
- UserLogger 266
- Username Authentication (Message Protection) 108

## V

- validation, via checkpoints 185
- variables, test 270
- version control 203
- views
  - Attachments 81
  - Data Queries 82
  - Data Query Properties 83
  - Events 80
  - General 68
  - HTTP 78
  - Input/Checkpoints 70
  - REST HTTP Input 78
  - Security Settings 82
  - SOAP fault 76
  - Test variables 77
- virtual machine settings 66
- VM settings 66
- VuGen integration
  - adding steps 47
  - calling a script 68
  - UFT testing 36

## W

- W3ES security 142
- warnings 94
- WCF Custom Binding 84, 88
- WCF Federation 90
- WCF HTTP Binding 91
- WCF services 103



- Web Services
  - Federation security 105
  - message signatures 102
  - negative testing 114
  - security scenario 100
  - testing 23
  - transport JMS 36
- Windows Authentication 107
- Windows store 138
- workbench 24
- WS-Addressing version 145
- WSE3 security 105
- WsHttpBinding 106
- WS-Reliable Messaging 109
- WS-Security 110

## **X**

- X.509 certificates
  - details 112
  - identities 104
  - requiring 143
- XML data grid 171

