OPTIMIZE

MERCURY QUICKTEST PROFESSIONAL™

WEB SERVICES ADD-IN

VERSION 9.2

Guide

MERCURY™
BUSINESS TECHNOLOGY OPTIMIZATION

Mercury QuickTest Professional Web Services Add-in

Guide

Version 9.2

Document Release Date: February 26, 2007

MERCURY™

Mercury QuickTest Professional Web Services Add-in Guide, Version 9.2

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

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Fax: (650) 603-5300
http://www.mercury.com

If you have any comments or suggestions regarding this document, please send them by e-mail to documentation@mercury.com.

QTPWSGD9.2/01

Table of Contents

PART I: SETTING UP WEB SERVICES ADD-IN SUPPORT

PART II: WORKING WITH THE WEB SERVICES ADD-IN

Welcome to This Guide

Welcome to the QuickTest Professional Web Services Add-in.

Web services are self-contained, modular, dynamic applications that can be described, published, located, or invoked over the network to create products, processes, and supply chains. They can be local, distributed, or Web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, HTML, and XML.

Web services usually provide a description to ensure that the client uses the data format expected by the server using a language known as *Web Services Definition Language* (*WSDL*).

The Web Services Add-in enables you to test your Web service using familiar QuickTest functionality, without the need for extensive knowledge of your Web service architecture. For example, you can use QuickTest Professional to invoke the operations of your Web service and verify returned XML data using special functionality that has been customized for Web services. You can use the specialized WebService test object operations to control the way in which QuickTest communicates with your service, including working with security, configuration, headers, and attachments. You can also use the Web Services Add-in together with any other QuickTest add-ins to create tests or components that test both the direct communications with your Web service and the front-end application that reflects the results of these communications.

The Web Service Testing Wizard enables you to automatically create a WebService test object and to generate steps to test the operations that your Web service supports based on a supplied WSDL file. When working with tests, the Web Service Testing Wizard also enables you to automatically generate checkpoints for these steps.

The Web Service Add Object Wizard enables you to automatically create a WebService test object. You can then add steps manually.

Using these Web service-specific features and operations helps to make Web service tests easier to read, maintain, enhance, and parameterize, enabling both advanced and novice users to create sophisticated tests on Web services.

Note: The Web Services Add-in is compliant with namespace and XPath standards.

For more information on XML standards, see http://www.w3.org/XML/

For more information on namespace standards, see http://www.w3.org/TR/1999/REC-xml-names-19990114/

For more information on XPath standards, see http://www.w3.org/TR/1999/REC-xpath-19991116

How This Guide Is Organized

This guide explains how to install the QuickTest Professional Web Services Add-in and use QuickTest Professional to test Web services.

This guide should be used in conjunction with the *QuickTest Professional User's Guide* and the *QuickTest Professional Object Model Reference*. These guides can be accessed online by choosing **Help** > **QuickTest Professional Help** from the QuickTest main window.

This guide contains:

Part I Setting Up Web Services Add-in Support

Details the process of installing and setting up the QuickTest Professional Web Services Add-in, including:

- Installing the Web Services Add-in
- Activating Web Services Support

Part II Working with the Web Services Add-in

Explains how to use the QuickTest Professional Web Services Add-in, including:

- Testing Web Services
- Working with XML Data

Note: The information, examples, and screen captures in this guide focus specifically on working with QuickTest tests. However, much of the information applies equally to components. Information that is unique to using the Web Services Add-in with Business Process Testing is indicated as such.

Business components and scripted components are part of Mercury Business Process Testing, which utilizes a keyword-driven methodology for testing applications. For more information, refer to the *QuickTest Professional User's Guide* and the *QuickTest Professional for Business Process Testing User's Guide*.

Who Should Read This Guide

This guide is intended for QuickTest Professional users at all levels who want to use the QuickTest Professional Web Services Add-in to test Web service applications.

Readers should already have some understanding of functional testing concepts and processes, and know which aspects of their application they want to test.

Note: The Web Services Add-in takes advantage of commonly used QuickTest features such as the object repository, Keyword View, and checkpoints and output value steps to enable you to test your Web service. You should have at least a basic understanding of these concepts before you begin working with the QuickTest Professional Web Services Add-in.

QuickTest Professional Online Documentation

QuickTest Professional includes the following online documentation:

Readme provides the latest news and information about QuickTest. Choose **Start** > **Programs** > **QuickTest Professional** > **Readme**.

QuickTest Professional Installation Guide explains how to install and set up QuickTest. Choose **Help** > **Printer-Friendly Documentation** > **Mercury QuickTest Professional Installation Guide**.

QuickTest Professional Tutorial teaches you basic QuickTest skills and shows you how to design tests for your applications. Choose **Help** > **QuickTest Professional Tutorial**.

Product Feature Movies provide an overview and step-by-step instructions describing how to use selected QuickTest features. Choose **Help** > **Product Feature Movies**.

Printer-Friendly Documentation displays the complete documentation set in Adobe portable document format (PDF). Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site (http://www.adobe.com). Choose **Help** > **Printer-Friendly Documentation**.

QuickTest Professional Help includes:

- **What's New in QuickTest Professional** describes the newest features, enhancements, and supported environments in the latest version of QuickTest.
- **QuickTest User's Guide** describes how to use QuickTest to test your application.
- **QuickTest for Business Process Testing User's Guide** provides step-by-step instructions for using QuickTest to create and manage assets for use with Business Process Testing.
- **QuickTest Object Model** describes QuickTest test objects, lists the methods and properties associated with each object, and provides syntax information and examples for each method and property.
- **QuickTest Advanced References** contains documentation for the following QuickTest COM and XML references:
    - **QuickTest Automation** provides syntax, descriptive information, and examples for the automation objects, methods, and properties. It also contains a detailed overview to help you get started writing QuickTest automation scripts. The automation object model assists you in automating test management, by providing objects, methods and properties that enable you to control virtually every QuickTest feature and capability.
    - **QuickTest Test Results Schema** documents the XML schema that enables you to customize your test results.
    - **QuickTest Test Object Schema** documents the XML schema that enables you to extend test object support in different environments.
    - **QuickTest Object Repository Automation** documents the Object Repository automation object model that enables you to manipulate QuickTest object repositories and their contents from outside of QuickTest.

- **VBScript Reference** contains Microsoft VBScript documentation, including VBScript, Script Runtime, and Windows Script Host.

To access the QuickTest Professional Help, choose **Help** > **QuickTest Professional Help**. You can also access the QuickTest Professional Help by clicking in selected QuickTest windows and dialog boxes and pressing F1. Additionally, you can view a description, syntax, and examples for a QuickTest test object, method, or property by placing the cursor on it and pressing F1.

Note: Your QuickTest Help may contain additional items relevant to any QuickTest add-ins you have installed. For more information, refer to the relevant add-in documentation.

Additional Online Resources

Knowledge Base uses your default Web browser to open the Mercury Customer Support Web site directly to the Knowledge Base landing page. Choose **Help** > **Knowledge Base**. The URL for this Web site is http://support.mercury.com/cgi-bin/portal/CSO/kbBrowse.jsp.

Customer Support Web Site uses your default Web browser to open the Mercury Customer Support Web site. This site enables you to browse the Mercury Support Knowledge Base and add your own articles. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help** > **Customer Support Web Site**. The URL for this Web site is http://support.mercury.com.

Send Feedback enables you to send online feedback about **QuickTest Professional** to the product team. Choose **Help** > **Send Feedback**.

Mercury Home Page uses your default Web browser to access Mercury's Web site. This site provides you with the most up-to-date information on Mercury and its products. This includes new software releases, seminars and trade shows, customer support, educational services, and more. Choose **Help** > **Mercury Home Page** and click the **Mercury Home** link in the top right corner of the page. The URL for this Web site is http://www.mercury.com.

Mercury Best Practices contain guidelines for planning, creating, deploying, and managing a world-class IT environment. Mercury provides three types of best practices: Process Best Practices, Product Best Practices, and People Best Practices. Licensed customers of Mercury software can read and use the Mercury Best Practices available from the Customer Support site, http://support.mercury.com.

Documentation Updates

Mercury is continually updating its product documentation with new information. You can download the latest version of this document from the Customer Support Web site (http://support.mercury.com).

To download updated documentation:

1 If you are not already logged into the Customer Support Web site, enter your login information and click **Login**.

2 Click the **Documentation** link.

3 Under **Please Select Product**, select **QuickTest Professional**.

Note that if the required product does not appear in the list, you must add it to your customer profile. Click **My Account** to update your profile.

4 Click **Retrieve**. The Documentation page opens and lists the documentation available for the current release and for previous releases. If a document was updated recently, **Updated** appears next to the document name.

5 Click a document link to download the documentation.

Typographical Conventions

This guide uses the following typographical conventions:

UI Elements and **Function Names** — This style indicates the names of interface elements on which you perform actions, file names or paths, and other items that require emphasis. For example, “Click the **Save** button.” It also indicates method or function names. For example, "The **wait_window** statement has the following parameters:"

Arguments — This style indicates method, property, or function arguments and book titles. For example, “Refer to the *Mercury User’s Guide*.”

<**Replace Value**> — Angle brackets enclose a part of a file path or URL address that should be replaced with an actual value. For example, <**MyProduct installation folder**>**bin**.

Example — This style is used for examples and text that is to be typed literally. For example, “Type Hello in the edit box.”

CTRL+C — This style indicates keyboard keys. For example, “Press ENTER.”

[ ] — Square brackets enclose optional arguments.

{ } — Curly brackets indicate that one of the enclosed values must be assigned to the current argument.

... — In a line of syntax, an ellipsis indicates that more items of the same format may be included. In a programming example, an ellipsis is used to indicate lines of a program that were intentionally omitted.

| — A vertical bar indicates that one of the options separated by the bar should be selected.

Part I

Setting Up Web Services Add-in Support

1

Installing the Web Services Add-in

This chapter lists the pre-installation requirements and explains how to install the Web Services Add-in.

Before You Install

Before you begin to install the Web Services Add-in, review the system requirements listed below.

To work successfully with the Web Services Add-in, your system configuration should meet the requirements as specified for QuickTest Professional 9.2 (in the *QuickTest Professional 9.2 Readme*), plus the following add-in specific requirements:

Prerequisites:	QuickTest Professional 9.2.
Operating System	Windows XP—Service Pack 2, or later, operating system.
Hard Disk Space:	85 MB of free hard disk space is required on the disk on which you want to install the Web Services Add-in, in addition to the space required for the QuickTest Professional installation. In addition, the disk on which Windows is installed (system disk) requires 110 MB of free hard disk space while the Web Services Add-in is being installed. After the installation is complete, 25 MB of this disk space is released (meaning that 85 MB of the system disk remains in use by QuickTest Professional).

Running the Setup Program

The Setup program installs add-in support in your QuickTest Professional installation folder for testing Web services.

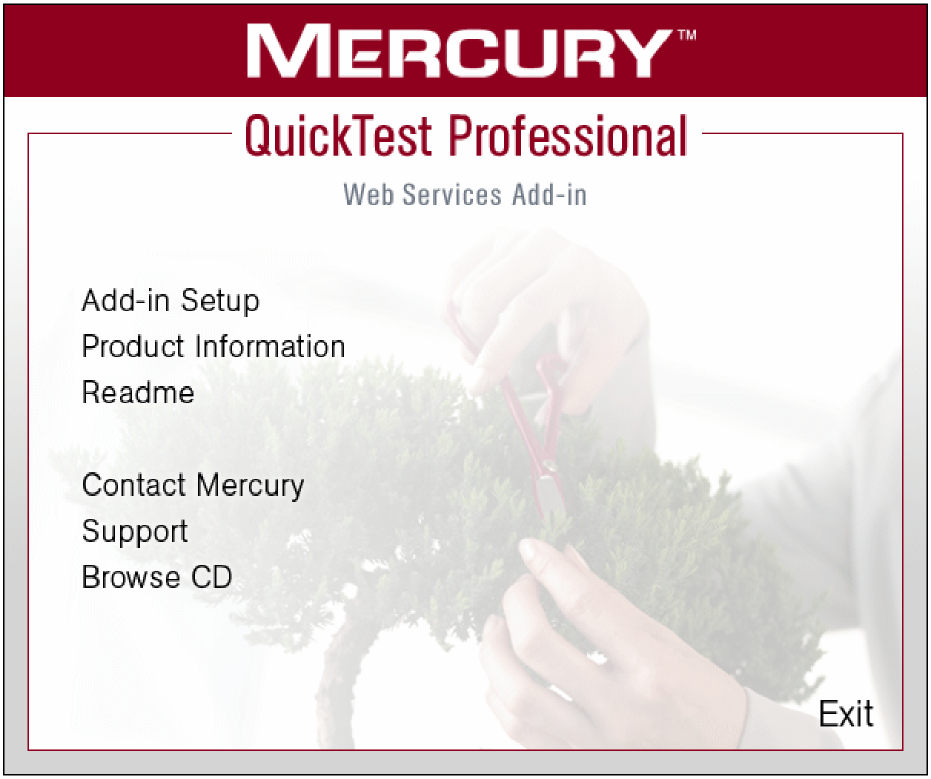
Notes:

- To install the Web Services Add-in, you must be logged on with administrator privileges.
- To install the QuickTest Professional Web Services Add-in silently (in the background), you must already have Microsoft WSE version 3.0 installed before you record or run the silent installation process. Silent installation is not supported for Windows XP 64-Bit Edition.
- You must not run any other installation at the same time as you run the Web Services Add-in installation.

To install the Web Services Add-in:

1. Close any instances of QuickTest Professional. It is also recommended to close all other open applications.
2. Insert the CD-ROM into the CD-ROM drive.
   - If the CD-ROM drive is on your local computer, the QuickTest Professional Web Services Add-in Setup window opens.

- If you are installing from a network drive, browse to it and double-click **autorun.exe** in the root folder of the CD-ROM. The QuickTest Professional Web Services Add-in Setup window opens.



The QuickTest Professional Web Services Add-in Setup window contains the following options:

- **Add-in Setup.** Starts the Web Services Add-in installation program.
- **Product Information.** Opens the product information site (http://www.mercury.com/us/products/quality-center/functional-testing/quicktest-professional/).
- **Readme.** Opens the *QuickTest Professional Web Services Add-in Readme* file.
- **Contact Mercury.** Opens the contact information page on the Mercury Web site (http://www.mercury.com/us/company/corporate-info/contact-us/).

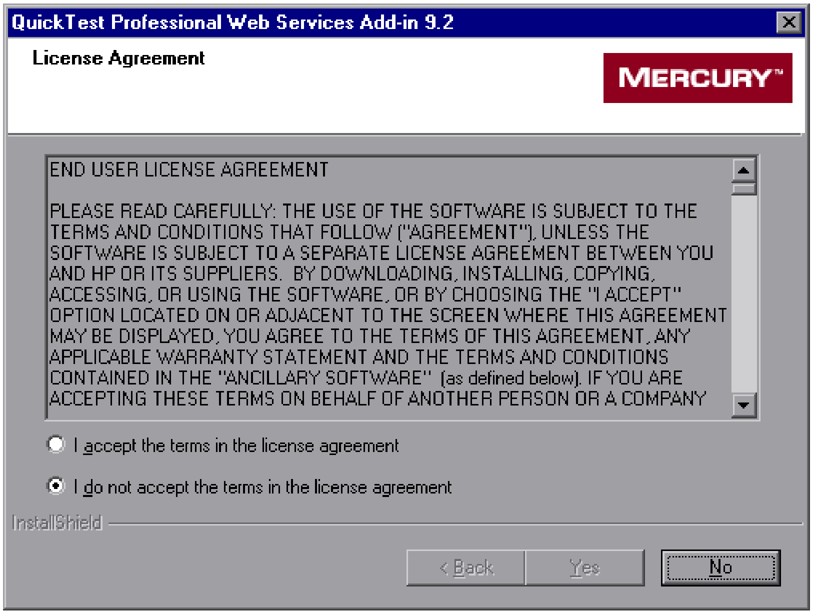
- **Support.** Opens the Mercury Customer Support Web site (http://support.mercury.com).
- **Browse CD.** Displays the contents of the QuickTest Professional Web Services Add-in CD-ROM.
- **Exit.** Exits the Setup program.

3. To start the Web Services Add-in Setup program, click **Add-in Setup**. The Web Services Add-in Setup program starts.
4. Microsoft WSE is required for the QuickTest Professional Web Services Add-in. If it is not installed on your computer a message box opens, enabling you to install it as part of the setup process. Click **Yes** in the message box to open the Microsoft WSE installation wizard.

Note: If you click **No**, the installation closes and the Web Services Add-in is not installed.

Follow the Microsoft WSE installation wizard steps to complete the WSE installation.

5 The License Agreement screen opens. Read the agreement.



QuickTest Professional Web Services Add-in 9.2

License Agreement

END USER LICENSE AGREEMENT

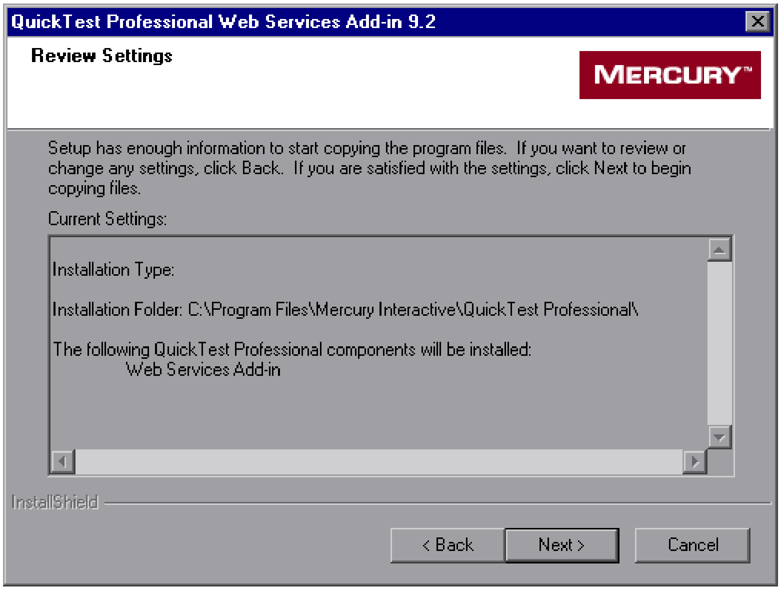
PLEASE READ CAREFULLY: THE USE OF THE SOFTWARE IS SUBJECT TO THE TERMS AND CONDITIONS THAT FOLLOW ("AGREEMENT"), UNLESS THE SOFTWARE IS SUBJECT TO A SEPARATE LICENSE AGREEMENT BETWEEN YOU AND HP OR ITS SUPPLIERS. BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE SOFTWARE, OR BY CHOOSING THE "I ACCEPT" OPTION LOCATED ON OR ADJACENT TO THE SCREEN WHERE THIS AGREEMENT MAY BE DISPLAYED, YOU AGREE TO THE TERMS OF THIS AGREEMENT, ANY APPLICABLE WARRANTY STATEMENT AND THE TERMS AND CONDITIONS CONTAINED IN THE "ANCILLARY SOFTWARE" (as defined below). IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY

○ I accept the terms in the license agreement

● I do not accept the terms in the license agreement

To install QuickTest Professional add-in support for testing Web services, you must accept the terms of the license agreement by selecting the **I accept the terms in the license agreement** option and clicking **Yes**. If you click **No**, the Setup program closes.

6 In the Review Settings screen, review the installation settings.



Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

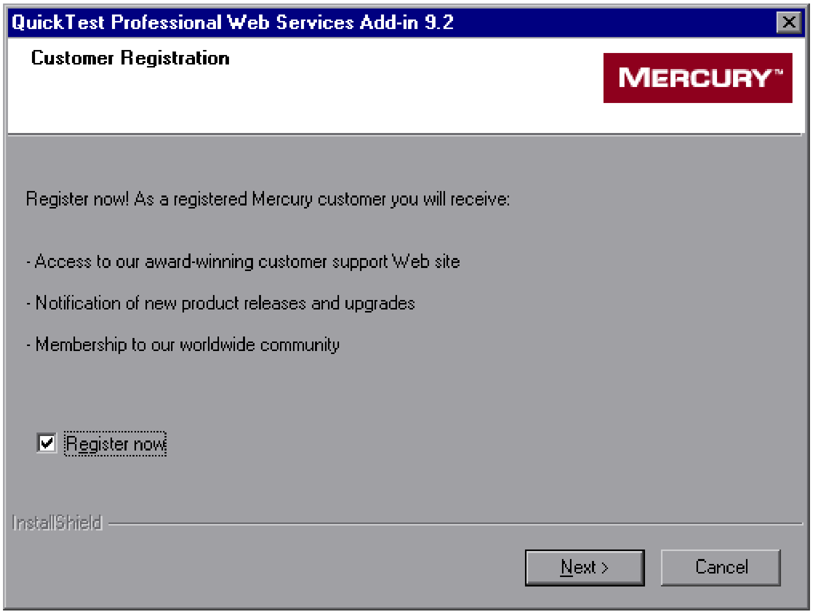
Installation Type:

Installation Folder: C:\Program Files\Mercury Interactive\QuickTest Professional\

The following QuickTest Professional components will be installed:
Web Services Add-in

Click **Next**. The installation process begins.

7 In the Customer Registration screen, you can specify whether to register your copy of the QuickTest Professional Web Services Add-in.



QuickTest Professional Web Services Add-in 9.2

Customer Registration

Register now! As a registered Mercury customer you will receive:

- Access to our award-winning customer support Web site

- Notification of new product releases and upgrades

- Membership to our worldwide community

☑ Register now

If you register, you receive:

- access to the Mercury award-winning Customer Support Web site.
- notification of new product releases and upgrades.
- membership to the Mercury worldwide community of testers.

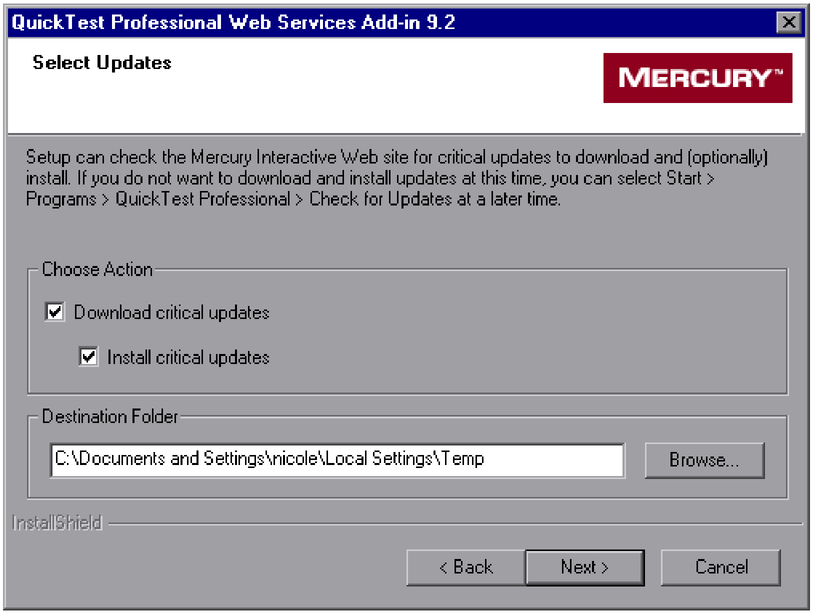
To register now, confirm that the **Register now** check box is selected. Click **Next**. Your browser opens to the Mercury Customer Support site (http://support.mercury.com). Follow the directions on the Customer Support site.

To register at a later time, or if you are already a registered user, clear the **Register now** check box and click **Next** to proceed with the installation.

8 The installation process checks whether there are any critical updates to be installed for the version of the Web Services Add-in you are installing. It also checks whether there are any critical updates to be installed for your version of QuickTest Professional and any other external add-ins that are installed on your computer.

If no critical updates are found, the installation continues with step 10.

If the Setup program finds any critical updates to be installed, the Select Updates screen opens.



QuickTest Professional Web Services Add-in 9.2

Select Updates

Setup can check the Mercury Interactive Web site for critical updates to download and (optionally) install. If you do not want to download and install updates at this time, you can select Start > Programs > QuickTest Professional > Check for Updates at a later time.

Choose Action

☑ Download critical updates

☑ Install critical updates

Destination Folder

C:\Documents and Settings\nicole\Local Settings\Temp

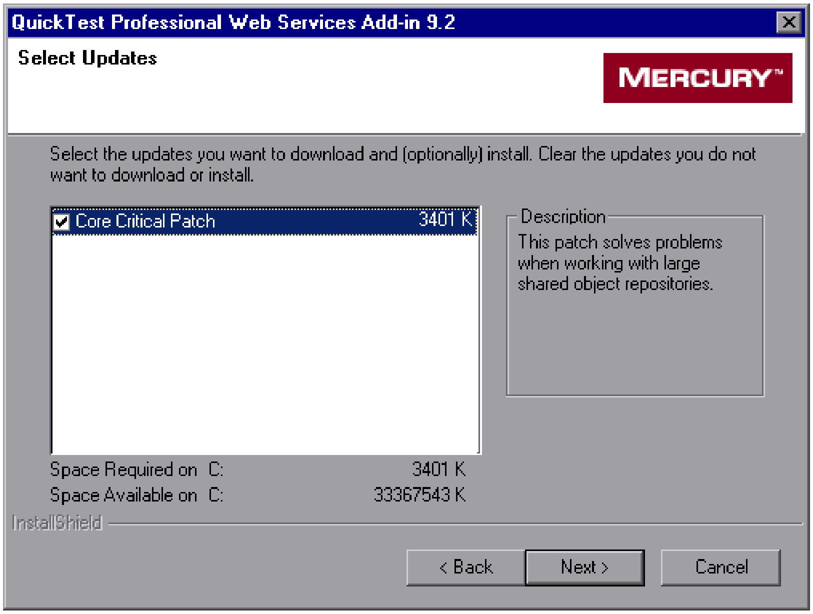
Choose one of the following options:

- ➤ **Download critical updates.** Downloads any critical updates to the specified destination folder. This option is selected by default.
- ➤ **Install critical updates.** Installs any downloaded critical updates. This option is selected by default.

Note: If the Setup program finds any critical updates to be installed, it is highly recommended to install them immediately. (You can also check for updates at a later time by choosing **Start** > **Programs** > **QuickTest Professional** > **Check for Updates**.)

A default download folder is displayed in the **Destination Folder** box. To select a different location to which to download the file, click **Browse**, choose a folder, and click **OK**.

Click **Next** to proceed. If you chose to download and (optionally) install critical updates, a list of all available critical updates opens. Otherwise, continue with step 10.



QuickTest Professional Web Services Add-in 9.2

Select Updates

Select the updates you want to download and (optionally) install. Clear the updates you do not want to download or install.

☑ Core Critical Patch 3401 K

Description
This patch solves problems when working with large shared object repositories.

Space Required on C: 3401 K
Space Available on C: 33367543 K

9 Select the updates that you want to download and (optionally) install. Clear the updates that you do not want to download or install.

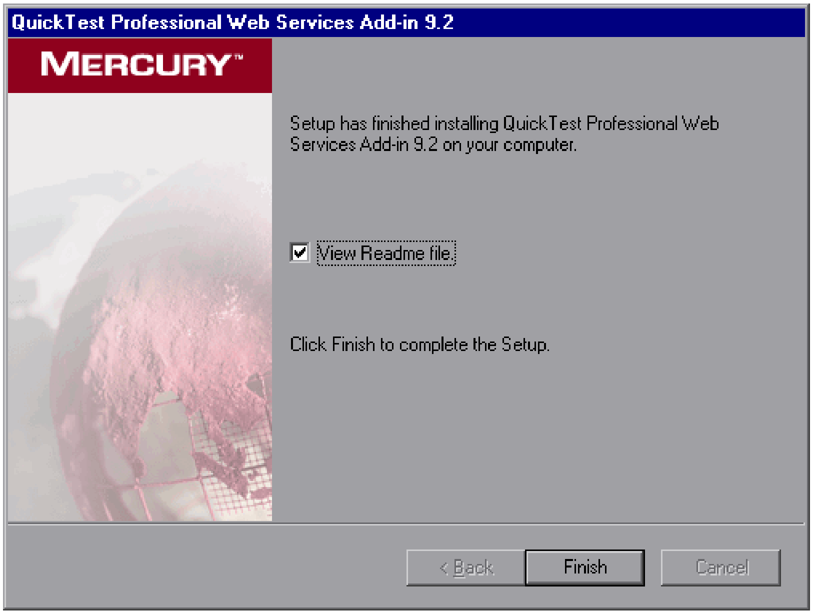
Click **Next** to proceed. QuickTest downloads the specified critical updates, according to the option you selected in step 8. If you chose to download and install the updates, they are installed at the end of the add-in installation process, after you restart your computer. (Note that you are always prompted to restart your computer if you choose to download and install critical updates.)

10 If the screen prompts you to restart your computer, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: If you are prompted to restart, it is strongly recommended that you do so as soon as possible. Delaying the system restart could result in unexpected system behavior.

If you are not prompted to restart, proceed to step 11. Otherwise, step 11 occurs after the restart.

11 The Setup Complete screen opens. If you want to open the *QuickTest Professional Web Services Add-in Readme* file at the end of the Setup process, select the **View Readme file** check box.



QuickTest Professional Web Services Add-in 9.2

Setup has finished installing QuickTest Professional Web Services Add-in 9.2 on your computer.

☑ View Readme file.

Click Finish to complete the Setup.

Click **Finish** to complete the Setup program.

In the Readme file, you can view the latest technical and troubleshooting information for the Web Services Add-in. To open the Readme file at another time, choose **Start** > **Programs** > **QuickTest Professional** > **Add-ins** > **Web Services Readme**.

2

Activating Web Services Support

Before you can work with the QuickTest Professional Web Services Add-in, you must make sure that the Web Services Add-in is loaded. You can load QuickTest without add-in support for Web services if you do not want to test Web services applications. You can also repair or uninstall your Web services installation. Your QuickTest Professional license enables all QuickTest features including the use of all QuickTest add-ins.

Loading QuickTest with Web Services Add-in Support

You use the Add-in Manager to load support for testing Web services applications.

Notes:

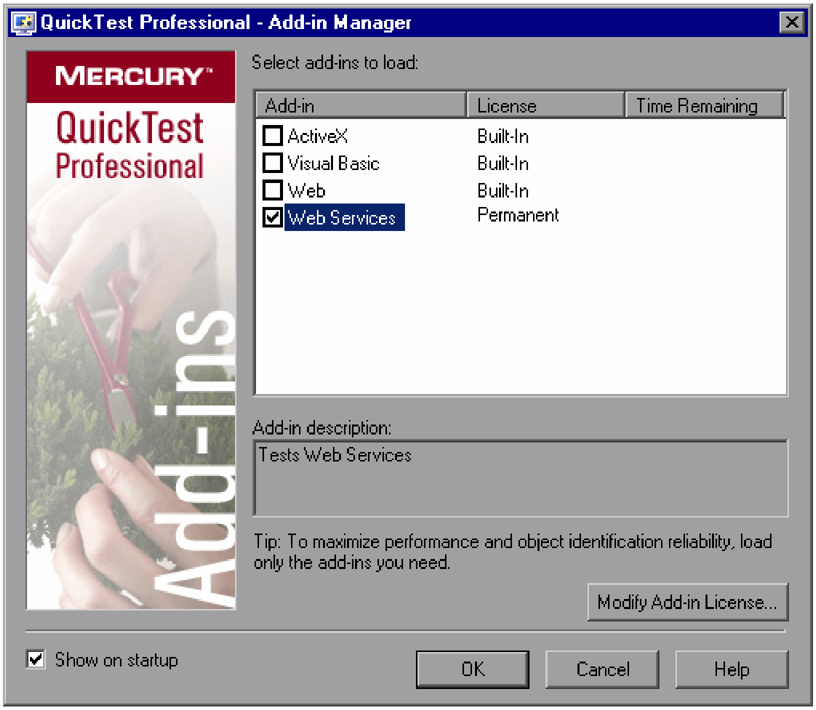
- For optimal performance when testing Web Services Add-in applications, it is strongly recommended that you work with *only* the QuickTest Professional Web Services Add-in loaded. Note that the QuickTest Professional Web Add-in does not need to be loaded when using the Web Services Add-in.
- When testing applications that do not contain Web services objects, it is strongly recommended that you do not load the Web Services Add-in.

To start QuickTest with add-in support for testing Web services applications:

1 Choose **Start** > **Programs** > **QuickTest Professional** > **QuickTest Professional**. The QuickTest Professional Add-in Manager dialog box opens.

(If the Add-in Manager dialog box does not open, see “Displaying the Add-in Manager Dialog Box” on page 30.)

2 In the add-in list, select **Web Services**.



Select add-ins to load:

Add-in	License	Time Remaining
□ ActiveX	Built-In	
□ Visual Basic	Built-In	
□ Web	Built-In	
☑ Web Services	Permanent	

Add-in description:

Tests Web Services

Tip: To maximize performance and object identification reliability, load only the add-ins you need.

☑ Show on startup

3 Click **OK**.

For more information on the Add-in Manager dialog box, refer to the *QuickTest Professional User's Guide*.

Displaying the Add-in Manager Dialog Box

You can set an option in QuickTest to determine whether the Add-in Manager opens when you open QuickTest, or whether it automatically loads the same add-ins that were loaded in the previous QuickTest session.

To instruct the Add-in Manager dialog box to open when you open QuickTest:

1. Choose **Start** > **Programs** > **QuickTest Professional** > **QuickTest Professional** to start QuickTest.
2. From the QuickTest menu, choose **Tools** > **Options** and click the **General** tab.
3. Select **Display Add-in Manager on startup**.
4. Click **OK** to close the Options dialog box.
5. Close and reopen QuickTest.

Loading QuickTest without Web Services Add-in Support

If you want to work with QuickTest without support for Web services, you can load QuickTest without the Web Services Add-in.

To load QuickTest without add-in support for Web services:

1. Choose **Start** > **Programs** > **QuickTest Professional** > **QuickTest Professional**. The QuickTest Professional Add-in Manager dialog box opens. (If the Add-in Manager dialog box does not open, see “Displaying the Add-in Manager Dialog Box” above.)
2. Clear the **Web Services** check box and click **OK**. QuickTest opens without add-in support for Web services.

Working with Function Libraries

QuickTest provides out-of-the-box resource function libraries. These files are specially designed to provide additional functionality for business components when working with Business Process Testing to test your Web services application using the QuickTest Professional Web Services Add-in. This enables you or Subject Matter Experts creating business process tests in Quality Center to choose functions that perform complex operations. For example, these files contain functions that enable you to open a specific application, write results of specific component steps to the test results, add steps to perform verification and output value operations (similar to QuickTest checkpoints and output values), and use keywords to perform a variety of other commonly performed operations. These function libraries are installed locally on your computer when you install QuickTest and the QuickTest Professional Web Services Add-in. To use these functions in your business components, you attach both the **Web_Services.txt** and **Common.txt** files to your Quality Center project and then associate them with your component's application area. It is recommended not to modify these function libraries, as QuickTest may overwrite them at a later stage, for example, when you upgrade QuickTest or the QuickTest Professional Web Services Add-in. For more information on working with application areas and business components, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

In addition, if you are working with tests but prefer to work in the Keyword View, you can associate the **Web_Services.txt** function library with your test. This function library contains functions that enable you to insert keywords into your test steps that are similar to the operations available for test steps in the Expert View.

The **Web_Services.txt** function library is located in your <**QuickTest installation folder**>**\dat\BPT_Resources** folder. For information on associating this function library with your test, refer to the *QuickTest Professional User's Guide*. For information on associating this function library with your component's application area, see "Attaching Function Libraries to Your Quality Center Project" on page 32 and "Associating Add-in Function Libraries with Your Application Area" on page 34. For more information on working with the functions in this function library, see "Working with Web Service Operations" on page 77.

Attaching Function Libraries to Your Quality Center Project

Before you can use the functions stored in the **Web_Services.txt** and **Common.txt** function libraries in your component steps, you must attach both of these files to your Quality Center project.

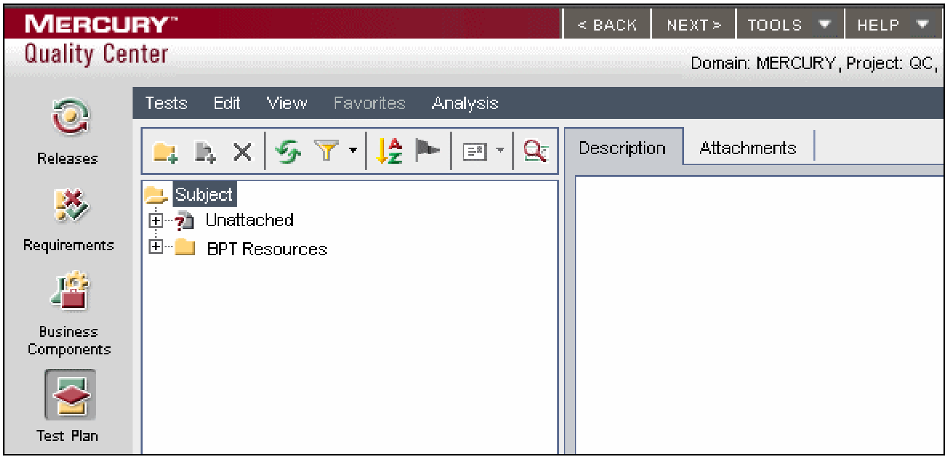
Note: To attach function libraries to your Quality Center project, you must have the permissions required to add attachments to folders in the Test Plan module of your project. By default, these permissions are assigned to the **TDAdmin**, **Project Manager**, **QATester**, and **Developer** user groups. For more information, contact your Quality Center administrator.

By default, the first time that you connect to a Quality Center project with Business Process Testing support, QuickTest sets up default Business Process Testing folders and files (including **Web_Services.txt** and **Common.txt**) in your Quality Center project. If for any reason one of these files does not appear in the list (for example, if the file was inadvertently deleted), you can attach it manually, as described in this section.

Note: You can also attach and use function libraries that contain user-defined functions. For more information, refer to the *QuickTest Professional User's Guide*.

To add a function library to your Quality Center project:

1 Open the Test Plan module in a Quality Center project that has Business Process Testing support. For more information on opening Quality Center projects, refer to the *Mercury Quality Center User's Guide*.



2 Expand the **BPT Resources** folder in the test plan tree and select **Libraries**.

3 Click the **Attachments** tab in the right pane. The Attachments tab opens and displays the function libraries that are attached to your project.

4 Click the **File** button and browse to the QuickTest Professional installation folder on your local computer, expand the **dat\BPT_Resources** folder, and select the missing function library. Click **Open**. The relevant function library is added to the Attachments list.

5 In the **Description** box, add text that clearly describes what the function library can be used for, for example: Default function library containing Web services-related operations that can be used when creating business components.

6 If needed, repeat steps 3 to 5 to attach any other function library to your Quality Center project.

You can now associate the function library with your application area or component.

Note: By default, all Business Process Testing-related function libraries are stored in the **BPT Resources\Libraries** folder in the test plan tree. You can store function libraries in any convenient location in the test plan tree, as long as you associate the function library with the relevant application area, as described below.

Associating Add-in Function Libraries with Your Application Area

After you attach the **Web_Services.txt** (and **Common.txt**, if not already attached) function libraries to your Quality Center project, you can associate both of these resource files with an application area. This enables you or Subject Matter Experts working in Quality Center to use the functions included in these files when creating, modifying, or running component steps. When you associate these resource files with an application area, all associated components can access their functions.

To associate add-in function libraries with an application area:

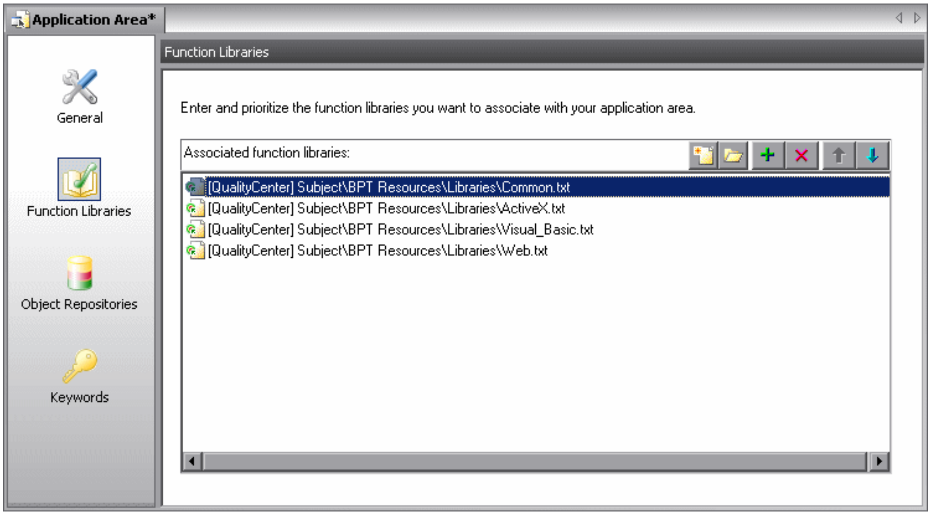
1. Verify that the **Web_Services.txt** and **Common.txt** files are attached to your Quality Center project.

   Tip: You can check that these files are attached to your project by opening the Test Plan module of your Quality Center project, clicking the **Attachments** tab, and expanding the folders to display the files stored in **Subject\BPT Resources\Libraries**. If the **Common.txt** and/or the **Web_Services.txt** files are not displayed, you can add them to your project, as described in "Attaching Function Libraries to Your Quality Center Project" on page 32.

2. Open QuickTest and connect to your Quality Center project with Business Process Testing support. For information on connecting to a Quality Center project, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

3 Open the application area on which business components that test Web services applications will be based or create a new application area (**File** > **Open** > **Application Area** or **File** > **New** > **Application Area**). For more information on opening and creating application areas and associating business components, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

4 Click **Function Libraries** in the sidebar. The list of function libraries currently associated with the application area is displayed in the Function Libraries pane.



Enter and prioritize the function libraries you want to associate with your application area.

Associated function libraries:

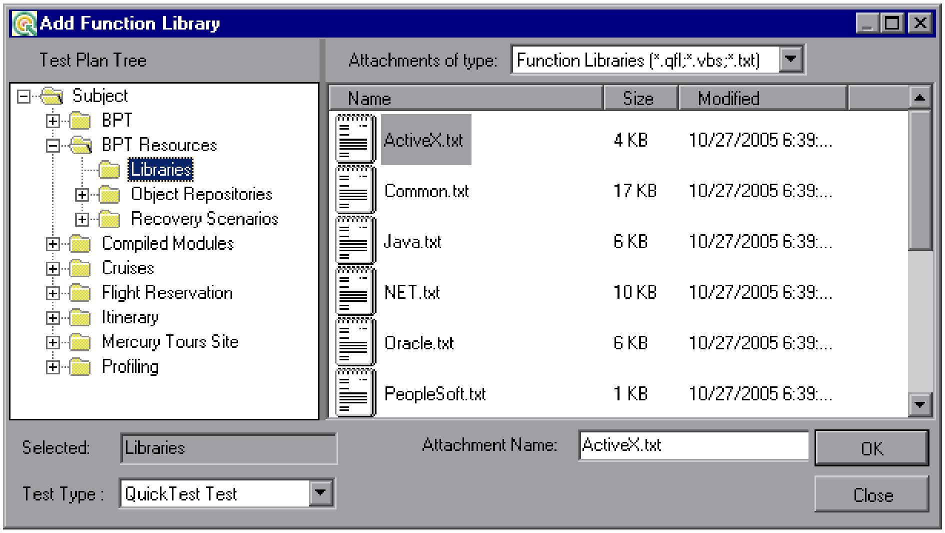
[QualityCenter] Subject\BPT Resources\Libraries\Common.txt
[QualityCenter] Subject\BPT Resources\Libraries\ActiveX.txt
[QualityCenter] Subject\BPT Resources\Libraries\Visual_Basic.txt
[QualityCenter] Subject\BPT Resources\Libraries\Web.txt

5 Click the **Add** button. QuickTest adds a blank line as well as a browse button enabling you to locate a function library in a Quality Center project.

Note: You can also type the entire Quality Center path manually. If you do, you must include [QualityCenter]<space> at the beginning of the path, for example: [QualityCenter] Subject\BPT Resources.

---

6 Click the browse button. The Add Function Library dialog box opens.



Attachments of type: Function Libraries (*.qfl;*.vbs;*.txt)

Name	Size	Modified
ActiveX.txt	4 KB	10/27/2005 6:39:...
Common.txt	17 KB	10/27/2005 6:39:...
Java.txt	6 KB	10/27/2005 6:39:...
NET.txt	10 KB	10/27/2005 6:39:...
Oracle.txt	6 KB	10/27/2005 6:39:...
PeopleSoft.txt	1 KB	10/27/2005 6:39:...

Selected: Libraries

Test Type : QuickTest Test

Attachment Name: ActiveX.txt

7 Browse to the **Subject\BPT Resources\Libraries** folder, select the **Web_Services.txt** file, and click **OK**. The file is added to the **Associated function libraries** area in the Function Libraries pane of the application area. (For any component associated with this application area, the file is also added to the Resources tab of the Business Component Settings dialog box in read-only format.)

8 If the **Common.txt** file is not displayed in the **Associated function libraries** area, repeat steps 5-7 and associate it with your application area.

You can now view and use the functions located in **Web_Services.txt** and **Common.txt** function libraries from any component that references these resource files from its associated application area.

Repairing and Uninstalling the Web Services Add-in

You can repair a QuickTest Professional Web Services Add-in installation that has become corrupted. In addition, you can uninstall the QuickTest Professional Web Services Add-in without uninstalling QuickTest or any other add-ins.

Note: You can also uninstall the QuickTest Professional application entirely. If you uninstall QuickTest Professional, the uninstall program removes all installed features, including any external add-ins that are installed. For more information on uninstalling QuickTest Professional, refer to the *QuickTest Professional Installation Guide*.

Repairing the Web Services Add-in Installation

Your QuickTest Professional Web Services Add-in 9.2 CD-ROM enables you to repair an existing Web Services Add-in 9.2 installation by replacing any missing or damaged files in your current Web Services Add-in installation.

To repair your Web Services Add-in installation:

1 Insert the QuickTest Professional Web Services Add-in CD-ROM into the CD-ROM drive. If the CD-ROM drive is on your local computer, the QuickTest Professional Web Services Add-in Setup window opens.

If the CD-ROM is in a network drive, double-click **autorun.exe** in the root folder of the CD-ROM. The QuickTest Professional Web Services Add-in Setup window opens.

Note: You must use the CD-ROM with the exact same version of the Web Services Add-in that you used for the original installation.

---

2 Click **Add-in Setup**. The Welcome screen opens.

3. In the Welcome screen, select **Repair** and click **Next**.

   The maintenance program repairs your QuickTest Professional Web Services Add-in installation.

4. The Maintenance Complete screen may prompt you to restart your computer. If it does, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: You can save any open files, but you should restart your computer as soon as possible. Delaying the system restart could result in unexpected system behavior.

---

5. Click **Finish** to complete the repair process.

Uninstalling the Web Services Add-in

You can uninstall the Web Services Add-in using either the **Add/Remove Programs** option in the Windows Control Panel, or using the QuickTest Professional Web Services Add-in CD-ROM.

To uninstall the Web Services Add-in using the Add/Remove Programs option:

1. Make sure that QuickTest is closed.
2. Choose **Start** > **Settings** > **Control Panel**.
3. Double-click the **Add/Remove Programs** option.
4. In the Add/Remove Programs dialog box, select **QuickTest Professional Web Services Add-in** and then click **Change/Remove**.
5. A message prompts you to confirm your decision to uninstall the QuickTest Professional Web Services Add-in. Click **Yes** to uninstall the Web Services Add-in. The uninstall program removes the Web Services Add-in from your computer. QuickTest Professional and any other installed add-ins remain on your computer.

Note: Clicking **No** keeps the Web Services Add-in installed on your computer.

---

6 The Maintenance Complete screen may prompt you to restart your computer. If it does, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: You can save any open files, but you should restart your computer as soon as possible. Delaying the system restart could result in unexpected system behavior.

---

7 Click **Finish** to complete the uninstall process.

To uninstall the Web Services Add-in using the QuickTest Professional Web Services Add-in CD-ROM:

1 Insert the QuickTest Professional Web Services Add-in CD-ROM into the CD-ROM drive. If the CD-ROM drive is on your local computer, the QuickTest Professional Web Services Add-in Setup window opens.

If the CD-ROM is in a network drive, double-click **autorun.exe** in the root folder of the CD-ROM. The QuickTest Professional Web Services Add-in Setup window opens.

Note: You must use the CD-ROM with the exact same version of the Web Services Add-in that you used for the original installation.

---

2 Click **Add-in Setup**. The Welcome screen opens.

3 In the Welcome screen, select **Remove** and click **Next**.

4 A message prompts you to confirm your decision to uninstall the QuickTest Professional Web Services Add-in. Click **OK** to uninstall the Web Services Add-in. The uninstall program removes the Web Services Add-in from your computer. QuickTest Professional and any other installed add-ins remain on your computer.

Note: Clicking **Cancel** keeps the Web Services Add-in installed on your computer and returns to the previous screen.

5 The Maintenance Complete screen may prompt you to restart your computer. If it does, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: You can save any open files, but you should restart your computer as soon as possible. Delaying the system restart could result in unexpected system behavior.

6 Click **Finish** to complete the uninstall process.

Part II

Working with the Web Services Add-in

3

Testing Web Services

The QuickTest Professional Web Services Add-in supports the testing of Web services. In addition to the information in this guide, it is recommended to review the Web service-related articles in the Mercury Customer Support Knowledge Base (http://support.mercury.com/cgi-bin/portal/CSO/kbBrowse.jsp).

This chapter explains how to use QuickTest to create and run tests and components on Web services. For more information on working with QuickTest, refer to the *QuickTest Professional User's Guide*.

About Testing Web Services

QuickTest Professional supports keyword-driven testing on your Web Service. This means that after you add a WebService test object to your object repository, you can access all of the operations that the WebService supports from the **Operation** column of the Keyword View, the **Operation** box in the Step Generator, or by using IntelliSense in the Expert View. You can add a WebService test object using the Web Service Add Object Wizard (described on page 67) or the Web Service Testing Wizard. You can also use the Web Service Testing Wizard to automatically generate steps and checkpoints on the operations supported by your Web Service. You can enhance your test manually by entering operations in the Expert View. For more information on working with the Expert View, refer to the *QuickTest Professional User's Guide*.

When working with components in QuickTest, you can use the Web Service Testing Wizard to automatically generate steps. Additionally, QuickTest provides commonly used Web service functions in the **Web_Services.txt** function library. You can use these functions if the function library is associated with your component's application area. For more information, see "Working with Function Libraries" on page 31.

You (or an Automation Engineer) can wrap additional Web Service-specific functionality in other function libraries associated with the component's application area. For more information on working with function libraries and application areas, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

When working with multiple toolkits, it is recommended to create a separate test or component for use with each required toolkit. For information on specifying a toolkit for your test or component, see "Setting Web Services Test Options" on page 73 and "Defining Web Service Test or Component Settings" on page 76.

Note: If your Web Service is protected by a firewall, or your Web service requests are routed via a proxy server, you may need to insert a **SetProxy** statement to set the required proxy information for the service before the first step containing an operation on the corresponding WebService test object. For more information, refer to the **Web Services** section of the *QuickTest Professional Object Model Reference*. If you are working with components, you or an Automation Engineer may need to wrap this functionality in a user-defined function and associate the function library with your application area.

Understanding the Web Service Testing Wizard

The Web Service Testing Wizard helps you select the WSDL source, set security options, and specify the Web service, port, and operations you want to test. For tests, you can also choose to automatically insert checkpoints with the proper syntax for each selected operation. When you finish, the wizard creates a WebService test object that represents the Web Service and port you want to test and inserts the relevant steps directly into your test or component.

You can then update the generated steps of your test or component by replacing the generated argument values with valid values, updating the expected values, and selecting the nodes you want to check in your checkpoints (tests only). After you perform these steps, you can update the data by performing an update run (**Automation** > **Update Run Mode**). For more information, refer to the *QuickTest Professional User's Guide*. You can also enhance the generated test or component by inserting additional steps or, for tests, adding programming logic.

After you associate the **Web_Services.txt** function library with your component's application area, you can also perform verification operations similar to checkpoints, and you can insert a wide range of Web service security operations in your component steps. For more information, see "Working with Web Service Operations" on page 77.

Note: If you want to automatically add a WebService test object to your object repository, and then generate steps manually for the operations it supports, you can use the Web Service Add Object Wizard. For more information, see "Adding WebService Test Objects to the Object Repository" on page 67.

To open the Web Service Testing Wizard:

Click the **Web Services Wizard** button or choose **Automation** > **Web Service Testing Wizard**. The Web Service Testing Wizard opens.

The Web Service Testing Wizard includes the following screens:

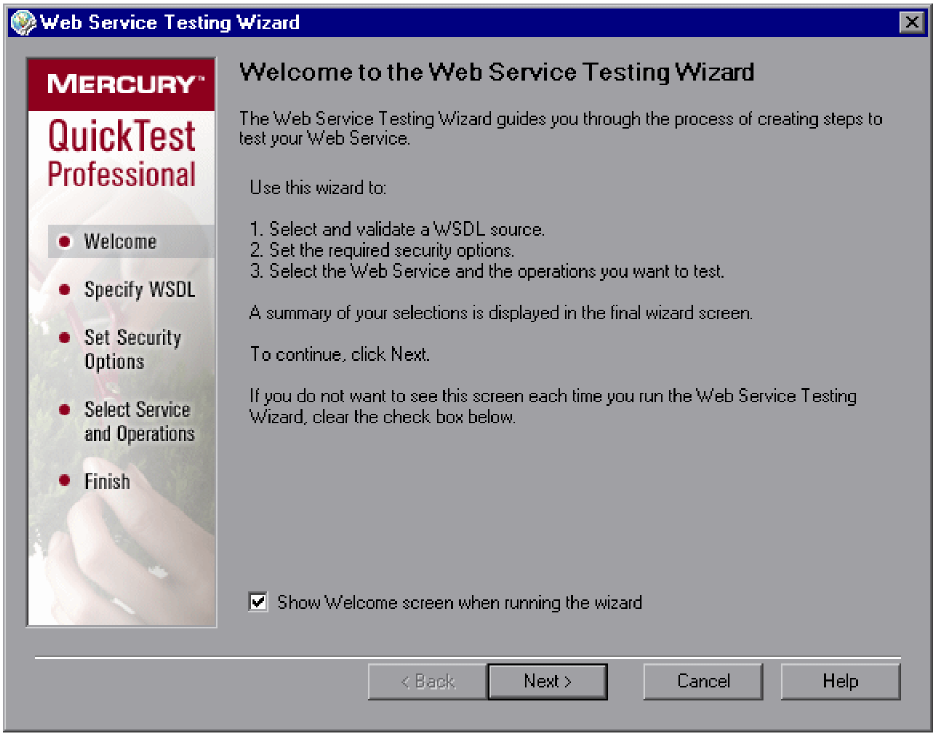
- Welcome to the Web Service Testing Wizard Screen. Provides an overview of the wizard steps.
- Web Service Testing Wizard - Specify WSDL for Scanning Screen. Enables you to select the source of the test object you want to include in the generated steps.
- Web Service Testing Wizard - Set Security Options Screen. Enables you to select the security tokens required for communication with the Web service you want to test, and to set their property values.
- Web Service Testing Wizard - Select Service and Operations Screen. Enables you to select the WSDL service you want to test and the service operations you want to include in your generated test.
- Web Service Testing Wizard - Summary Screen. Displays a summary of your selections and (for tests) enables you to choose whether to add checkpoints after each relevant operation step.

When you click **Finish**, the basic test is created. You complete it by entering relevant operation argument values and by setting checkpoint expected values and other preferences. For more information, see "Completing and Enhancing Your Generated Test" on page 62.

Note: Running your test or component without entering valid argument values may cause the test to fail.

Welcome to the Web Service Testing Wizard Screen

The Welcome to the Web Service Testing Wizard screen is the first screen displayed when you open the wizard. The screen describes the steps in the wizard.



Welcome to the Web Service Testing Wizard

The Web Service Testing Wizard guides you through the process of creating steps to test your Web Service.

Use this wizard to:

1. Select and validate a WSDL source.
2. Set the required security options.
3. Select the Web Service and the operations you want to test.

A summary of your selections is displayed in the final wizard screen.

To continue, click Next.

If you do not want to see this screen each time you run the Web Service Testing Wizard, clear the check box below.

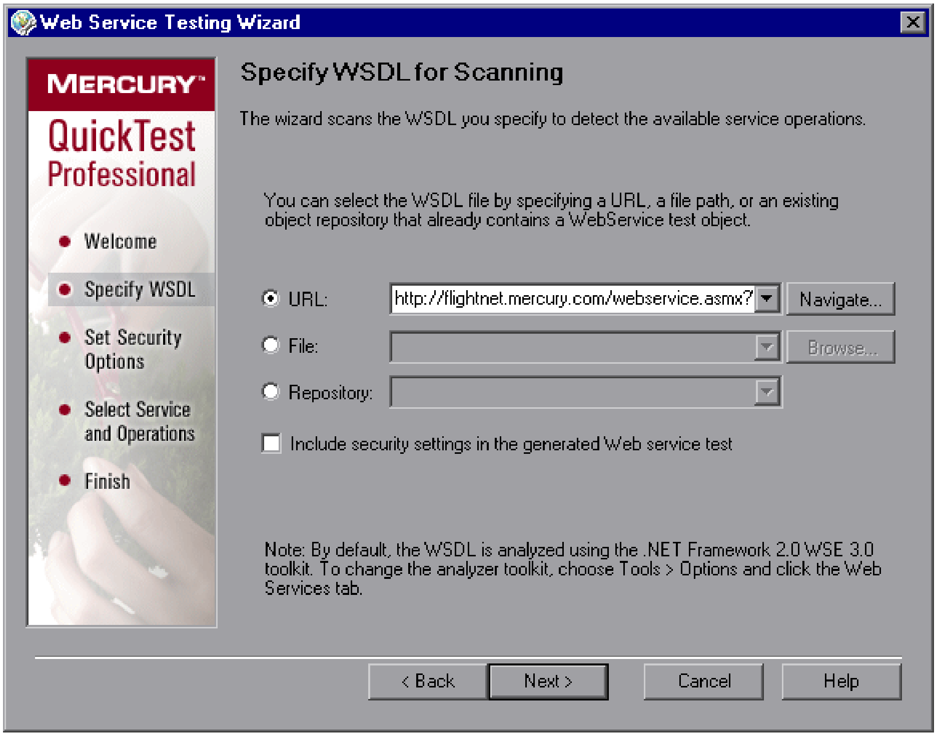
☑ Show Welcome screen when running the wizard

You can choose not to display the welcome screen when you open the wizard. Clear the **Show Welcome screen when running the wizard** check box in this screen, or choose **Tools** > **Options** in the QuickTest window and then clear **Show Welcome step in Web Service Testing Wizard** in the Web Services tab of the Options dialog box. For more information about the Web Services tab, see "Setting Web Services Test Options" on page 73.

Click **Next** to continue to the Web Service Testing Wizard - Specify WSDL for Scanning Screen.

Web Service Testing Wizard - Specify WSDL for Scanning Screen

The Specify WSDL for Scanning screen enables you to select the required WSDL source. The source can be a URL, a WSDL file, or an existing WebService test object from the object repository.



Web Service Testing Wizard

Specify WSDL for Scanning

The wizard scans the WSDL you specify to detect the available service operations.

You can select the WSDL file by specifying a URL, a file path, or an existing object repository that already contains a WebService test object.

● URL: http://flightnet.mercury.com/webservice.asmx?

○ File:

○ Repository:

□ Include security settings in the generated Web service test

Note: By default, the WSDL is analyzed using the .NET Framework 2.0 WSE 3.0 toolkit. To change the analyzer toolkit, choose Tools > Options and click the Web Services tab.

Notes:

- The first time you open the Web Service Testing Wizard, a sample URL is displayed in the URL box. On subsequent uses of the wizard, the Specify WSDL for Scanning screen opens with the same settings as those set in the previous wizard session.
- By default, the WSDL source is analyzed using the .NET Framework 2.0 WSE 3.0 toolkit. You can change the toolkit if required. For more information, see “Specifying the Web Services Toolkit” on page 72.

In the Specify WSDL for Scanning screen, specify the WSDL source and whether you want to include security settings in your Web service test, as follows:

Select a radio button according to the WSDL source you want to test. Then enter the source. You can click the down arrow ▼ next to each box to view and select recently used items.

- If you want to locate a URL source on a Web server, click the **Navigate** button next to the **URL** box to open Microsoft Internet Explorer. The button name changes to **Capture**. Navigate to the required URL (WSDL file). Minimize the browser and click **Capture** or close the browser. The URL address is automatically entered in the **URL** box.
- If you want to locate a WSDL file, click **Browse** next to the **File** box to open the Browse for WSDL File dialog box. Browse to the required file.

  If you are currently connected to a Quality Center project, you can toggle between the file system and the test plan tree for the Quality Center project by clicking the **File System** or **Quality Center** buttons.

Tip: From the **Attachments of type** list in the Browse for WSDL File dialog box (or **Files of type** list when choosing a file in the file system), you can choose to view only **.wsdl** files, only **.xml** files, or view all the files in the selected location.

- If you want to create steps for a service (and port) that has already been defined as a test object in one of the repositories associated with the current action or component (via its application area), select **Repository**, and then select the relevant test object. This option enables QuickTest to access the WebService object directly, without processing the WSDL file, thus saving processing time as QuickTest identifies the service and its operations. (This in turn enables the next screen to open faster.)

Select the **Include security settings in the generated Web service test** check box if you want to specify the security tokens that are required for communication with the Web service you want to test.

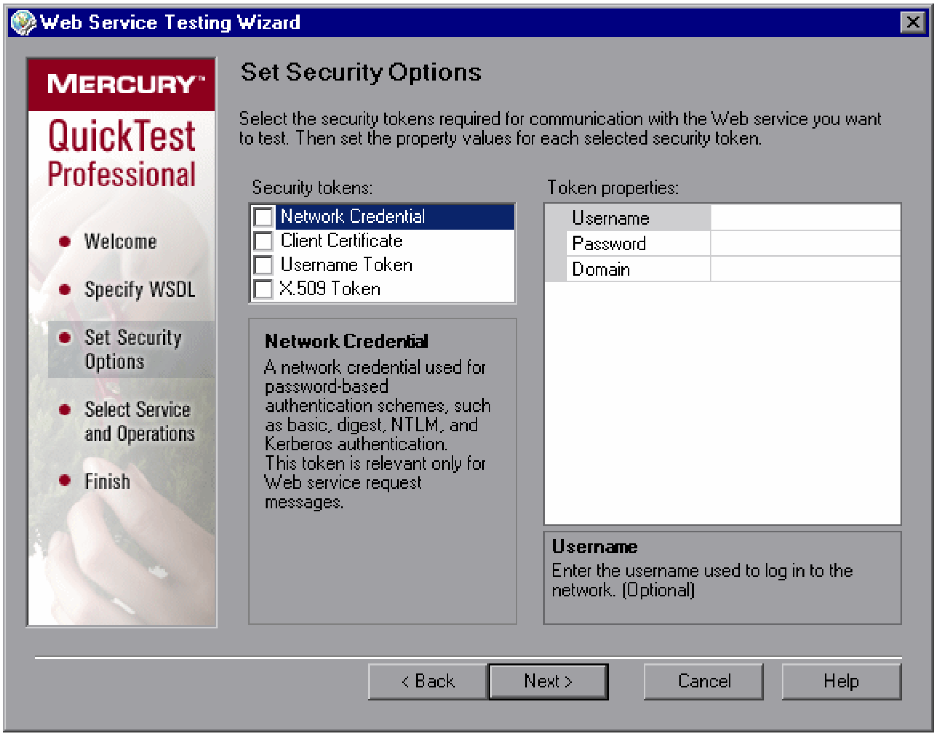
Click **Next**. Depending on your selections in the Specify WSDL for Scanning screen, one of the following occurs:

- If you specified a secure WSDL, the Network Credentials dialog box opens. Enter the login details required to access the WSDL and click **OK**. The Web Service Testing Wizard - Set Security Options Screen opens. The Network Credentials security token is selected and the token properties contain the credentials you specified. When QuickTest runs the test created by this wizard, the credentials saved with the test are used to access both the WSDL and the service.
- If you selected the **Include security settings in the generated Web service test** check box, the Web Service Testing Wizard - Set Security Options Screen opens.
- If you do not need to specify security settings, the Web Service Testing Wizard - Select Service and Operations Screen opens (described on page 58). Note that opening the next screen may take some time, depending upon the selected option. While scanning the WSDL, QuickTest displays a cyclic progress bar that runs until the scanning process is complete.

Note: The Set Security Options screen is available only for tests. Therefore, if you are working with components, when you click **Next**, the Web Service Testing Wizard - Select Service and Operations Screen opens (described on page 58). If required, you can insert steps containing security operations after you complete the wizard. These operations are available from the associated **Web_Services.txt** function library. For information on associating this function library with your component's application area, see "Working with Function Libraries" on page 31.

Web Service Testing Wizard - Set Security Options Screen

When working with tests, the Set Security Options screen enables you to specify the security tokens that are needed for communication with the Web service you want to test. For each selected security token, you also set its property values.



Set Security Options

Select the security tokens required for communication with the Web service you want to test. Then set the property values for each selected security token.

Security tokens:

- □ Network Credential
- □ Client Certificate
- □ Username Token
- □ X.509 Token

Token properties:

Username	
Password	
Domain	

Network Credential
A network credential used for password-based authentication schemes, such as basic, digest, NTLM, and Kerberos authentication.
This token is relevant only for Web service request messages.

Username
Enter the username used to log in to the network. (Optional)

After you complete the wizard, QuickTest converts your selections into one or more test steps, containing the corresponding operations and property values. For example, if you select **X.509 Token**, QuickTest might add steps similar to the following:

```
tokenID = WebService("FlightNetWebService").Security.AddX509Token
    (micRequestToken, XMLWarehouse("Certificate"))
```

Additionally, if you select **Client Certificate** or **X.509 Token** in the Security tokens area, the wizard loads the certificate that you specify in the Token properties area and saves it with the test. The wizard stores the raw data of the certificate as a binary string in a new XML structure. During a run session, QuickTest uses this certificate and does not load it from the external source that you specified. You can access the certificate from the XML Warehouse tab in the Settings dialog box (**File** > **Settings**). For more information on XML structures, see "Working with XML Structures" on page 91.

Tips:

- If you need to replace the certificate, run the wizard again to create a new XML structure containing the required certificate and add the relevant steps. Edit the test or component manually to remove the steps that used the old certificate.
- To instruct QuickTest to load the certificate from a file or a certificate store for each run session, manually create steps that use the **LoadX509CertificateFromFile** or **LoadX509CertificateFromStore** methods. These methods return an XMLData object that you can use as an argument for subsequent steps. For more information, refer to the **Web Services** section of the *QuickTest Professional Object Model Reference*.

In the **Security tokens** area, select the check boxes for the security tokens needed for communication with the Web service you want to test. You can highlight any token to view its description in the **Description** area. When you highlight a token, its properties are displayed in the **Token properties** area. (Note that highlighting the token does not select its check box.) You can select as many security tokens as needed.

The following security tokens are available:

Network Credential

A network credential used for password-based authentication schemes, such as basic, digest, NTLM, and Kerberos authentication. This token is relevant only for Web service request messages.

The **Network Credential** token includes the following properties:

- **Username.** The user name used to log in to the network.
- **Password.** The password used to log in to the network.
- **Domain.** The network domain name. (Optional)

Client Certificate

A client certificate used mostly when a client uses the SSL3.0/PCT1 protocol to connect to a server, and the server requires client certificates for mutual authentication.

The **Client Certificate** token includes the following properties:

- **StoreType.** Indicates whether the certificate is located in the file system or installed on the computer.
  Available options: **File** and **Store**.

  If you select **File**, the following **FileCertificate** properties are displayed:

  - **FileName.** The path of the file that contains the certificate data. Enter the file name or click the browse button to locate the certificate file.
  - **Password.** The password required to access the certificate file. (Optional)

If you select **Store**, the following **StoreCertificate** properties are displayed:

- **Location.** Indicates whether the certificate is installed for the current user or for anyone using the computer.
  Available options: **CurrentUser** and **LocalMachine**.
- **Store.** The store in which the certificate is located. QuickTest displays a list of standard locations for installed certificates. If the certificate that you need to use in the test is located elsewhere, use the **LoadX509CertificateFromStore** method. For more information refer to the **Web Services** section of the *QuickTest Professional Object Model Reference* (**Help** > **QuickTest Professional Help**).
- **Certificate.** The client certificate to use for authentication. To select a certificate, click the browse button. The Select Certificate dialog box opens, displaying all of the certificates installed in the store you specified. Select a certificate and click **OK**.

Username Token

Username and password security credentials.

The **Username** token includes the following properties:

- **TokenDirection.** Indicates whether the security token should be appended to all subsequent request messages or all subsequent response messages.
  Available options: **Request** and **Response**.
- **Username.** The user name used to sign or encrypt SOAP messages.
- **Password.** The password used to sign or encrypt SOAP messages.
- **SendMode.** Indicates how the password should be sent.
  Available options: **None**, **PlainText**, and **Hashed**.
- **ProtectionMode.** Indicates the protection mode to be applied to all subsequent Web service operations.
  Available options: **None**, **Signature**, **Encryption**, and **Both**.

X.509 Token

An X.509 certificate used for signing and/or encrypting Web service request messages when a server's public certificate is required.

The **X.509 token** includes the following properties:

- **TokenDirection.** Indicates whether the security token should be appended to all subsequent request messages or all subsequent response messages.
  Available options: **Request** and **Response**.
- **StoreType.** Indicates whether the certificate is located in the file system or installed on the computer.
  Available options: **File** and **Store**.

  If you select **File**, the following **FileCertificate** properties are displayed:

  - **FileName.** The path of the file that contains the certificate data. Enter the file name or click the browse button to locate the certificate file.
  - **Password.** The password required to access the certificate file. (Optional)

  If you select **Store**, the following **StoreCertificate** properties are displayed:

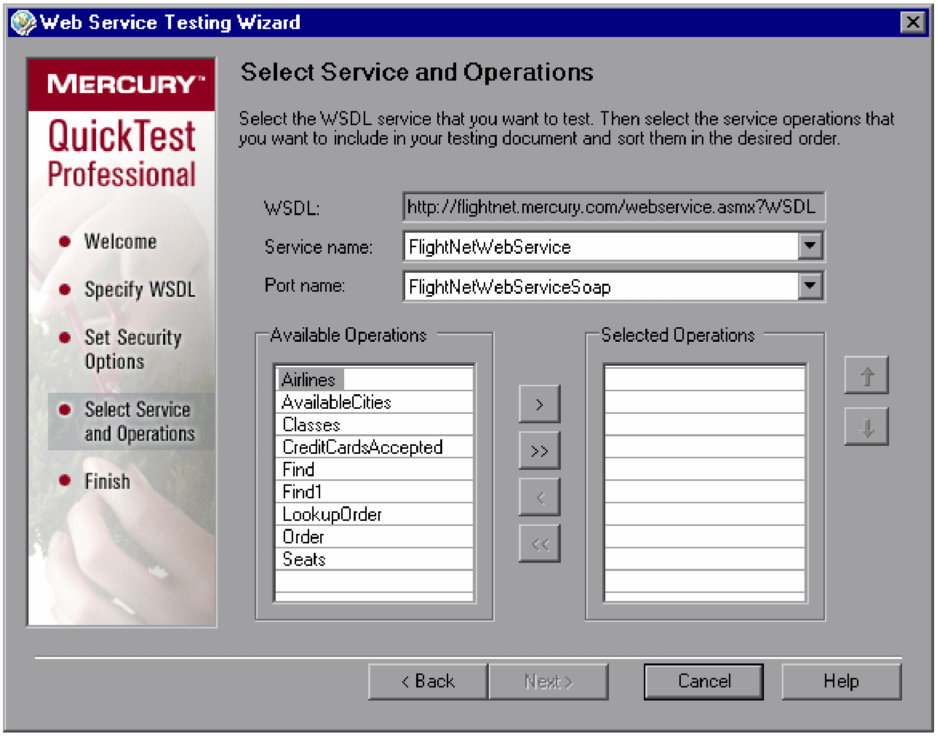
  - **Location.** Indicates whether the certificate is installed for the current user or for anyone using the computer.
    Available options: **CurrentUser** or **LocalMachine**.
  - **Store.** The store in which the certificate is located. QuickTest displays a list of standard locations for installed certificates. If the certificate that you need to use in the test is located elsewhere, use the **LoadX509CertificateFromStore** method. For more information refer to the **Web Services** section of the *QuickTest Professional Object Model Reference* (**Help** > **QuickTest Professional Help**).
  - **Certificate.** The client certificate to use for authentication. To select a certificate, click the browse button. The Select Certificate dialog box opens, displaying all of the certificates installed in the store you specified. Select a certificate and click **OK**.
- **ProtectionMode.** Indicates the protection mode to be applied to all subsequent Web service operations.
  Available options: **None**, **Signature**, **Encryption**, and **Both**.

Set the properties for each selected security token by highlighting the token in the **Security tokens** area and modifying its property values in the **Token properties** area. When you highlight a property, its description is displayed below the **Token properties** area.

Click **Next** to continue to the Web Service Testing Wizard **-** Select Service and Operations Screen.

Web Service Testing Wizard - Select Service and Operations Screen

The Select Service and Operations screen enables you to select a service and a port from the source WSDL you specified in the Specify WSDL screen (described on page 48). The test object that the wizard generates represents this specific service and port. You can then select the operations to be included in the test or component steps from the list of available operations supported by that Web service. You can also arrange them in the required testing order.



Select Service and Operations

Select the WSDL service that you want to test. Then select the service operations that you want to include in your testing document and sort them in the desired order.

WSDL: http://flightnet.mercury.com/webservice.asmx?WSDL

Service name: FlightNetWebService

Port name: FlightNetWebServiceSoap

Available Operations

- Airlines
- AvailableCities
- Classes
- CreditCardsAccepted
- Find
- Find1
- LookupOrder
- Order
- Seats

Selected Operations

From the **Service name** list and **Port name** list, select the service and port that you want to test. The name of the service you select is also used as the default name of the created test object. If a description of the Web service is available, it is displayed as a tooltip when the cursor is positioned over the service name.

Tip: You can rename a WebService test object in the Object Repository. Choose **Resources** > **Object Repository** to open the repository. Then select the test object, right-click, and choose **Rename** from the menu.

Note: The list of ports displays all of the ports in the selected service that work with a supported protocol. For a list of supported protocols, see the *QuickTest Professional Web Services Add-in Readme*.

In the **Available Operations** list, double-click each operation you want to include, or select an operation and click the arrow button > to add it to the **Selected Operations** list. You can add the same operation more than once.

To remove an operation from the **Selected Operations** list, you can double-click it in the **Selected Operations** list or select the operation and click the < arrow button.

Tip: Click the double arrow buttons (>> and <<) to move all the operations from one list to the other. Select multiple operations (using the SHIFT and/or CTRL keys on your keyboard) and click the arrow buttons (> and <) to move only the selected operations from one list to the other.

↑ ↓

Use the up and down arrows to sort the **Selected Operations** list into the required testing order.

Important Information for Microsoft .NET Framework Toolkit Users

If you are working with a .NET Framework toolkit, and one or more Web service operations in your WSDL require you to provide parameter data in a SOAP header, you can use the **SetHeaderField_<FieldName>Value** operation to do so. (**<FieldName>** represents the name of the .NET field in which the parameter data is stored and **Value** is appended to the field name.)

During a run session, this operation instructs QuickTest to save the header field value you specify, as if you are working directly with a .NET client. Thereafter, QuickTest inserts this header field value every time it sends a message requiring this header. If a response message updates this header field value, the updated header field value is saved and is used for all subsequent method calls.

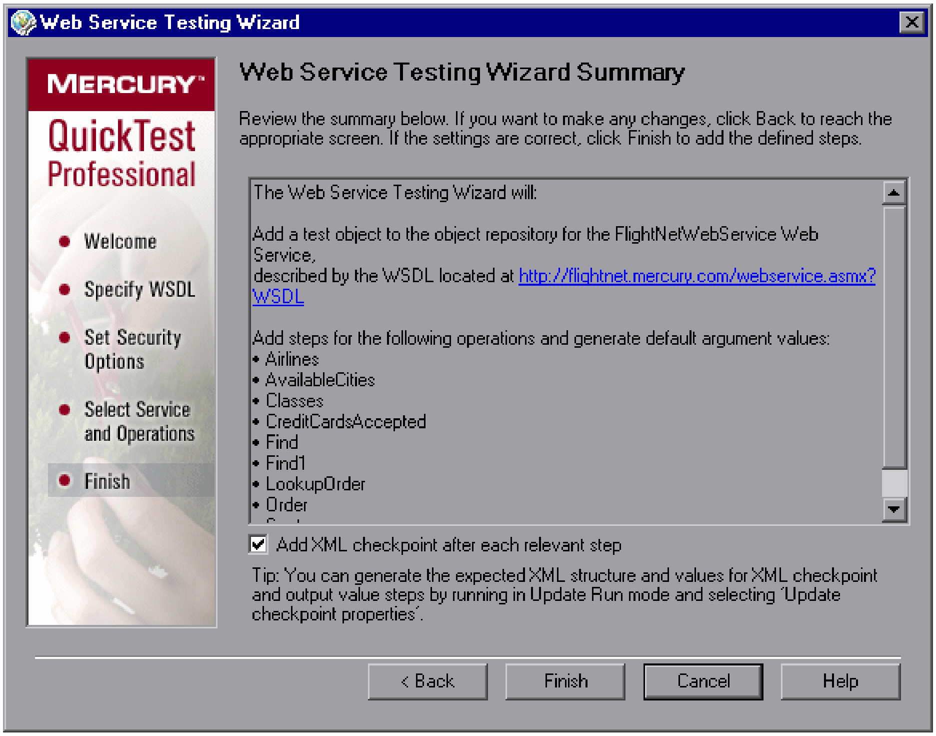
In the **Available Operations** list, select any operation beginning with **SetHeaderField_** and move it to the **Selected Operations** list. Then move this operation to the top of the **Selected Operations** list (or at least above any operations that require parameterized header data).

For more information on the **SetHeaderField_** operation, see “Accessing Operations Supported by the Web Service You Are Testing” on page 80.

Click **Next** to continue to the Web Service Testing Wizard - Summary Screen.

Web Service Testing Wizard - Summary Screen

The Web Service Testing Wizard Summary screen provides a summary of the operations for which the wizard will add steps according to your selections.



Web Service Testing Wizard Summary

Review the summary below. If you want to make any changes, click Back to reach the appropriate screen. If the settings are correct, click Finish to add the defined steps.

The Web Service Testing Wizard will:

Add a test object to the object repository for the FlightNetWebService Web Service,
described by the WSDL located at http://flightnet.mercury.com/webservice.asmx?WSDL

Add steps for the following operations and generate default argument values:
- Airlines
- AvailableCities
- Classes
- CreditCardsAccepted
- Find
- Find1
- LookupOrder
- Order

☑ Add XML checkpoint after each relevant step

Tip: You can generate the expected XML structure and values for XML checkpoint and output value steps by running in Update Run mode and selecting 'Update checkpoint properties'.

When working with tests, you can automatically insert XML checkpoints by selecting **Add XML checkpoint after each relevant step** (selected by default). Selecting this check box adds an XML checkpoint for each step in the test that has a return value or an output argument. For more information about XML checkpoints, refer to the *QuickTest Professional User's Guide*.

Review the summary. If the settings are correct, click **Finish.** The WebService test object is stored in the local object repository, and the defined steps are converted to the proper syntax and inserted into your test or component.

Completing and Enhancing Your Generated Test

The Web Service Testing Wizard accelerates the process of designing a basic test that checks the operations that your Web service supports.

For each operation, the wizard generates default argument values with the proper value types. For each automatically created checkpoint, the wizard generates a generic XML structure as a place holder for the expected XML return values. Before you can run your test or component, you must replace the default values with the appropriate values for your test.

Review each of the argument values and supply a value that is appropriate for the operation. For argument values of type XML, the wizard generates an XML structure. Use the **Configure the value** button in the Keyword View **Value** column to open the XML structure and edit the generated values within the structure. Alternatively, you can open the XML structure from the XML Warehouse tab in the Test Settings or Business Component Settings dialog box (**File** > **Settings**). For more information on XML structures, see "Working with XML Structures" on page 91. After you modify the required values, you need to update the data by performing an update run (**Automation** > **Update Run Mode**). For more information, refer to the *QuickTest Professional User's Guide*.

Open each automatically generated checkpoint, populate the XML tree with the expected return values, and select the items that you want to check. For more information, see "Checking XML" on page 88 and refer to the Checking XML chapter in the *QuickTest Professional User's Guide*.

You may want to output values returned by one step for use as input in another step. For more information, see "Outputting XML Values" on page 90 and refer to the XML Output Values section in the *QuickTest Professional User's Guide*.

You can also add steps using the test object methods that the QuickTest Professional Web Services Add-in supplies to test the behavior of your Web service. For more information, see "Using QuickTest WebService Test Object Methods and Properties" on page 78 and refer to the Web Services section of the *QuickTest Professional Object Model Reference*.

Checking that Your WSDL Meets WS-I Standards

You can instruct QuickTest to analyze a specific WSDL source to check that it conforms to WS-I standards and complies with the WS-I Basic Profile.

Note: QuickTest performs the validation using the WS-I validation tool, which is a third-party application that is not provided with QuickTest. You can download **Interoperability Testing Tools 1.1** from the Web Services Interoperability Organization Web site at http://www.ws-i.org. Note that it must be installed locally.

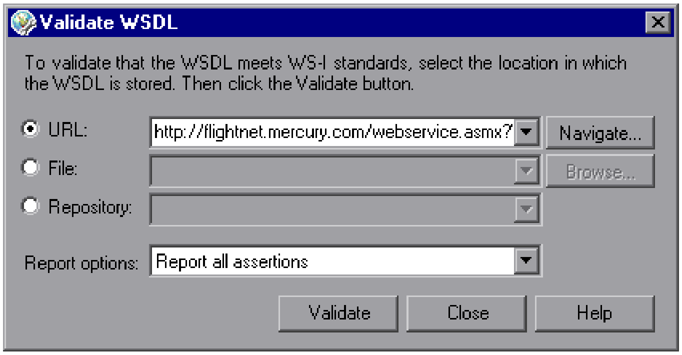
You can also validate that the WSDL source conforms to WS-I standards programmatically using either the **WebService.ValidateWSDL** method or the **WSUtil.ValidateWSDL** method. For more information, refer to the **Web Services** section of the *QuickTest Professional Object Model Reference*.

When the validation is complete, you can view the validation results, which indicate whether the file conforms to WS-I profile guidelines. Note that it is possible that your WebService tests or components on the specified WSDL will run successfully even if they do not conform to these guidelines.

Before you can use the WS-I validation tool, you need to specify the location of the WS-I validation tool. You do this in the Web Services tab of the Options dialog box (**Tools** > **Options**). For more information, see “Setting Web Services Test Options” on page 73.

To check that your WSDL meets WS-I standards:

1 Select **Tools** > **Validate WSDL** or press ALT+T+L. The Validate WSDL dialog box opens.



To validate that the WSDL meets WS-I standards, select the location in which the WSDL is stored. Then click the Validate button.

● URL: http://flightnet.mercury.com/webservice.asmx?

○ File:

○ Repository:

Report options: Report all assertions

2 Specify the location in which the WSDL is stored by selecting a radio button according to the WSDL source you want to test. Then enter the source. You can click the down arrow ▼ next to each box to view and select recently used items.

- ➤ To specify a WSDL from a URL source on a Web server, select **URL**. Enter the URL address manually or click the **Navigate** button next to the **URL** box to open your default browser. The button name changes to **Capture**. Navigate to the required URL. Minimize the browser and click **Capture** or close the browser. The URL address is automatically entered in the **URL** box.

- ➤ To specify a WSDL file, select **File**. Enter the file path manually or click **Browse** next to the **File** box to open the Browse for WSDL File dialog box. Browse to the required file.

  If you are currently connected to a Quality Center project, you can toggle between the file system and the test plan tree of the Quality Center project by clicking the **File System** or **Quality Center** buttons.

Tip: From the **Attachments of type** list in the Browse for WSDL File dialog box (or **Files of type** list when choosing a file in the file system), you can choose to view only **.wsdl** files, only **.xml** files, or view all the files in the selected location.

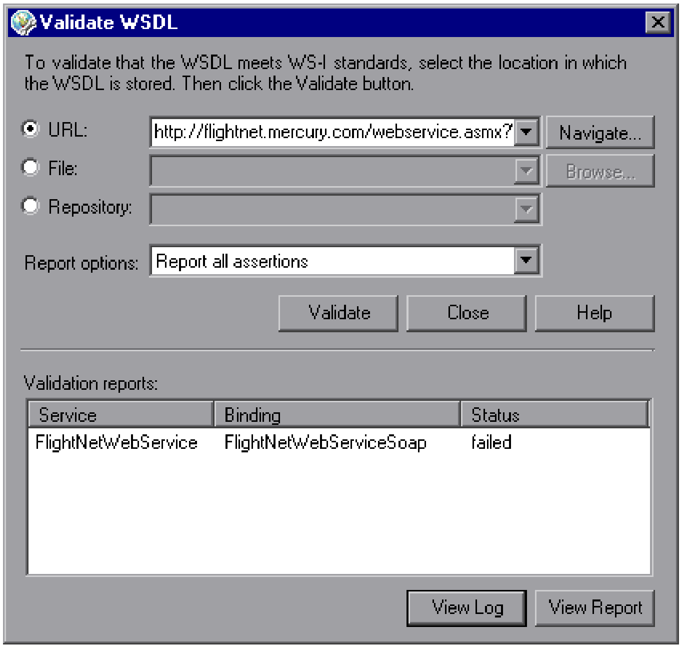
- To specify a WSDL that defines a service for which a test object has already been created in one of the repositories associated with the current action (or component), select **Repository**, and then select the relevant test object. QuickTest locates the WSDL according to the location specified in the **wsdl** property of the WebService test object.

3 Click the **Report options** down arrow and select the criteria for the assertion results to be included in the validation report. (Test assertions are used by the validation tool to analyze whether a Web service conforms to WS-I standards.)

- **Report all assertions.** Reports the results of all assertions.
- **Report all assertions but passed assertions.** Reports the results of all assertions except those that have a "passed" result.
- **Report only failed assertions.** Reports the results of assertions that have a "failed" result only.

4 Click the **Validate** button to analyze the specified WSDL source and check that it conforms to WS-I standards. The validation is performed for each binding defined in the WSDL. This process may take some time.

When the check is complete, the Validate WSDL dialog box expands to display the results of the validation check in the **Validation reports** area. This area lists the reports generated by the WS-I testing tool and indicates the status of each report.



Validate WSDL

To validate that the WSDL meets WS-I standards, select the location in which the WSDL is stored. Then click the Validate button.

● URL: http://flightnet.mercury.com/webservice.asmx?

○ File:

○ Repository:

Report options: Report all assertions

Validation reports:

Service	Binding	Status
FlightNetWebService	FlightNetWebServiceSoap	failed

5 Highlight a report in the **Validation reports** area:

- ➤ Click the **View Log** button to view the WS-I testing tool log for the selected report item.
- ➤ Click the **View Report** button to open the WS-I Profile Conformance Report for the selected report item in a Web browser.

Adding WebService Test Objects to the Object Repository

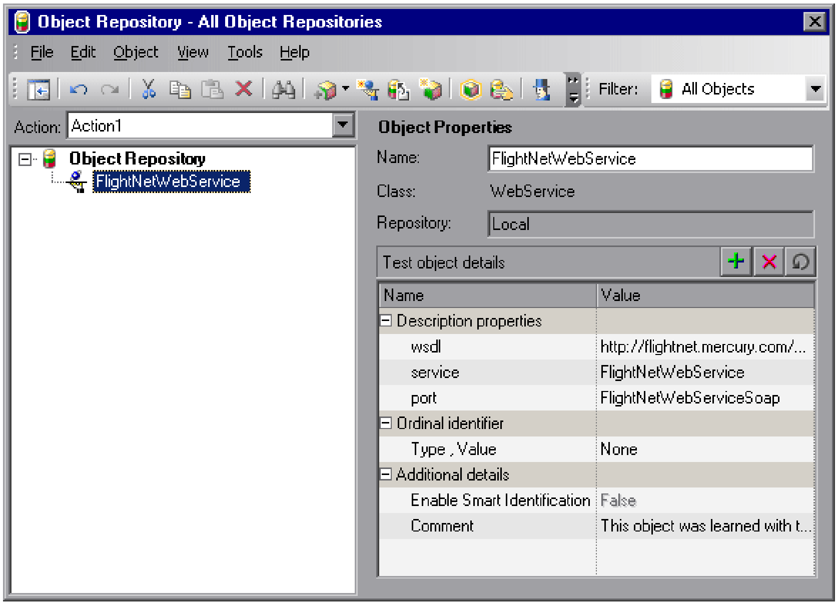
The Web Services Add-in enables you to add WebService test objects to your object repository using the Web Service Add Object Wizard. You can then use the new test object to add steps to your test or component. After a WebService test object is added to the object repository, you can access all operations supported by the Web service in the **Operations** column of the Keyword View, in the Expert View IntelliSense, or in the Step Generator. In IntelliSense and in the Step Generator, you can view the operations supported by the Web service in the upper part of the box. These are the operations that were derived from the WSDL source. QuickTest WebService test object operations are displayed in the lower part of the box.

The Web Service Add Object Wizard helps you select the WSDL source, the Web service, and the port you want to test.

Note: If you want to create a WebService test object and automatically generate steps for the operations it supports, you can use the Web Service Testing Wizard. For more information, see "Understanding the Web Service Testing Wizard" on page 45.

To open the Web Service Add Object Wizard:

1 Click the **Object Repository** toolbar button, or choose **Resources** > **Object Repository**. The Object Repository dialog box opens.


Object Repository - All Object Repositories
File Edit Object View Tools Help
Object Repository
Object Properties
Test object details

Filter: All Objects

Action: Action1

Object Repository
- FlightNetWebService

Name: FlightNetWebService

Class: WebService

Repository: Local

Name	Value
⊟ Description properties	
wsdl	http://flightnet.mercury.com/...
service	FlightNetWebService
port	FlightNetWebServiceSoap
⊟ Ordinal identifier	
Type , Value	None
⊟ Additional details	
Enable Smart Identification	False
Comment	This object was learned with t...

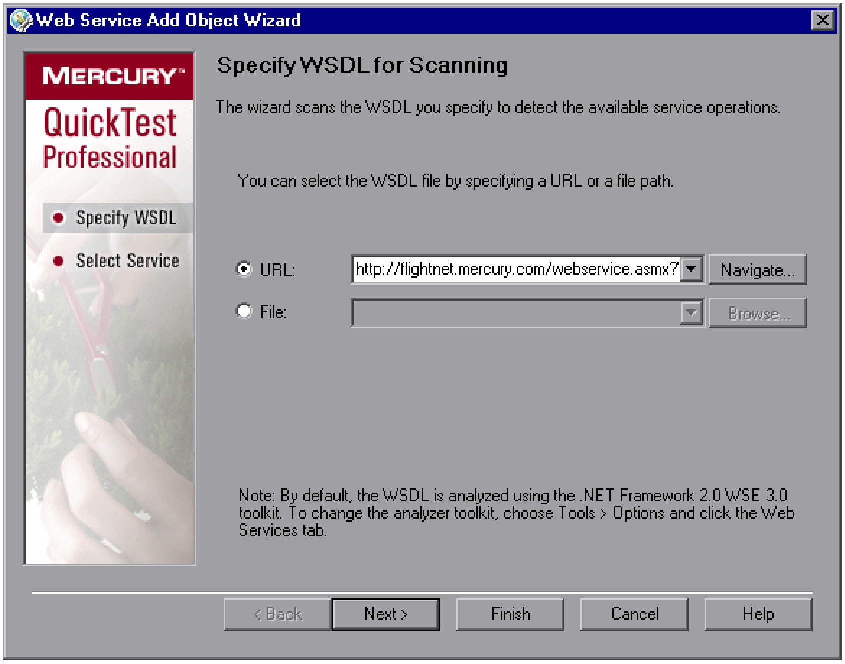
2 Click the **Web Service Add Object Wizard** button or select **Object** > **Web Service Add Object Wizard**. The Web Service Add Object Wizard opens to show the Specify WSDL for Scanning screen.

The Web Service Add Object Wizard includes the following screens:

- Add Object Wizard - Specify WSDL for Scanning Screen. Enables you to select and validate the source of the test object you want to create.
- Add Object Wizard - Select Service Screen. Enables you to select the WSDL service for which you want to create a test object.

Add Object Wizard - Specify WSDL for Scanning Screen

The Specify WSDL for Scanning screen is the first screen displayed when you open the Add Object wizard. This screen enables you to specify a URL or file as the WSDL source.



Specify WSDL for Scanning

The wizard scans the WSDL you specify to detect the available service operations.

You can select the WSDL file by specifying a URL or a file path.

● URL: http://flightnet.mercury.com/webservice.asmx?

○ File:

Note: By default, the WSDL is analyzed using the .NET Framework 2.0 WSE 3.0 toolkit. To change the analyzer toolkit, choose Tools > Options and click the Web Services tab.

Notes:

- The first time you open the Web Service Add Object Wizard, a sample URL is displayed in the URL box. On subsequent uses of the wizard, the Specify WSDL for Scanning screen opens with the same settings as those set in the previous wizard session.
- By default, the WSDL source is analyzed using the .NET Framework 2.0 WSE 3.0 toolkit. You can change the toolkit if required. For more information, see “Specifying the Web Services Toolkit” on page 72.

Select a radio button according to the WSDL source you want to test. Then enter the source. You can click the down arrow ▾ next to each box to view and select recently used items.

- If you want to locate a URL source on a Web server, click the **Navigate** button next to the **URL** box to open your default browser. The button name changes to **Capture**. Navigate to the required URL. Minimize the browser and click **Capture** or close the browser. The URL address is automatically entered in the **URL** box.
- If you want to locate a WSDL file, click **Browse** next to the **File** box to open the Browse for WSDL File dialog box. Browse to the required file.

  If you are currently connected to a Quality Center project, you can toggle between the file system and the test plan tree for the Quality Center project by clicking the **File System** or **Quality Center** buttons.

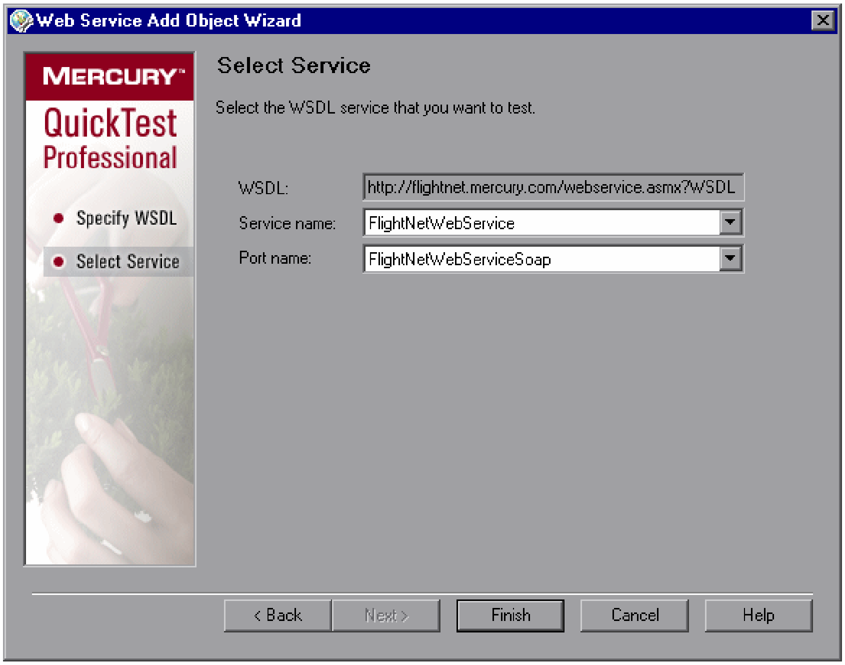
Tip: From the **Attachments of type** list in the Browse for WSDL File dialog box (or **Files of type** list when choosing a file in the file system), you can choose to view only **.wsdl** files, only **.xml** files, or view all the files in the selected location.

Click **Next** to continue to the Add Object Wizard - Select Service Screen or click **Finish** if you want the Wizard to create your test object using the first Web service and the first port described in the WSDL.

Note: If you specified a secure WSDL, the Network Credentials dialog box opens. Enter the login details required to access the WSDL and click **OK**.

Add Object Wizard - Select Service Screen

The Select Service screen in the Add Object wizard enables you to select the Web service and port that you want to test.



Select Service

Select the WSDL service that you want to test.

WSDL: http://flightnet.mercury.com/webservice.asmx?WSDL

Service name: FlightNetWebService

Port name: FlightNetWebServiceSoap

From the **Service name** list and the **Port name** list, select the service and port that you want to test and click **Finish**. The wizard adds a WebService test object to the object repository, representing the specific Web service and port. If a description of the Web service is available, it is displayed as a tooltip when the cursor is positioned over the service name.

Note: The list of ports displays all of the ports in the selected service that work with a supported protocol. For a list of supported protocols, see the *QuickTest Professional Web Services Add-in Readme*.

Close the Object Repository dialog box. You can now use the test object and its operations in the steps of your test.

Specifying the Web Services Toolkit

Before creating WebService test objects and steps, you can specify the Web services toolkit that you want QuickTest Professional to use when learning new test objects and when running your Web service steps. If you are working with multiple toolkits, it is recommended to create a unique test or component for each toolkit.

The toolkit you specify for learning Web service objects is a global setting used for all Web Service tests. You set this option in the Web Services tab of the Options dialog box (click the **Options** toolbar button, or choose **Tools** > **Options**). By default, QuickTest uses the Microsoft .NET Framework 2.0 WSE 3.0 toolkit when learning Web service objects and when running Web service tests and components. You can change the default, as described in "Setting Web Services Test Options" on page 73.

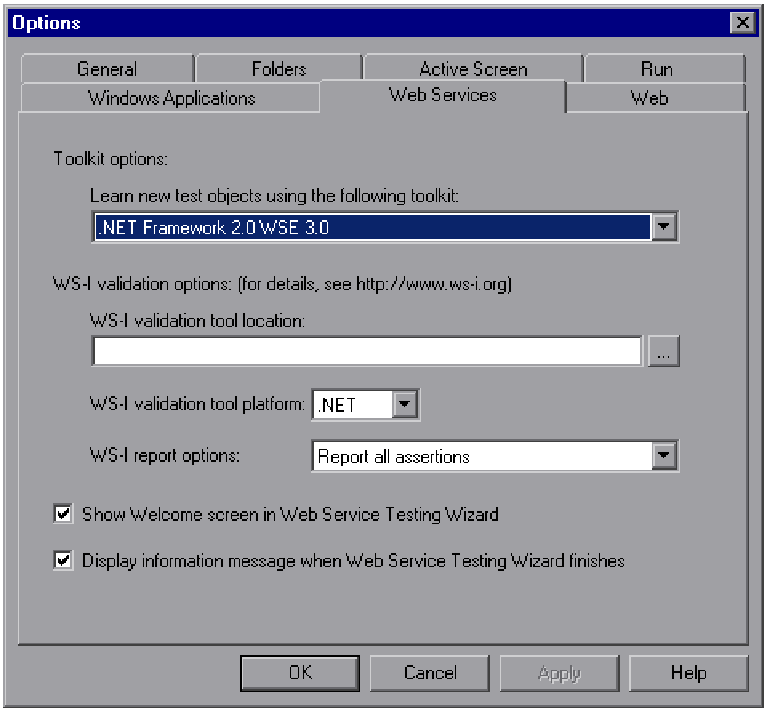
The toolkit you specify for running Web service tests and components is a local setting that is specific to a test or an application area. It is recommended to use the same toolkit you specify for learning Web service objects, otherwise your test may fail. For tests, you specify the toolkit in the Web Services tab of the Test Settings dialog box (click the **Settings** toolbar button, or choose **File** > **Settings**). For components, you specify the toolkit in the Web Services tab of the Application Area Settings dialog box (choose **File** > **Settings**, or click the **Settings** toolbar button, or click the **Additional Settings** button in the Application Area General pane).

Before you run a Web service test or component, make sure that the specified toolkit is installed on the computer on which QuickTest is installed. For example, if the test or component is set to run using the Microsoft .NET WSE 2.0 toolkit, make sure that this toolkit is installed on your QuickTest computer.

For more information, see "Defining Web Service Test or Component Settings" on page 76.

Setting Web Services Test Options

The Web Services options enable you to specify the toolkit that QuickTest uses when learning WebService test objects, specify WS-I validation preferences, and set additional display settings for the Web Service Testing Wizard.



Toolkit options:

Learn new test objects using the following toolkit: .NET Framework 2.0 WSE 3.0

WS-I validation options: (for details, see http://www.ws-i.org)

WS-I validation tool location:

WS-I validation tool platform: .NET

WS-I report options: Report all assertions

☑ Show Welcome screen in Web Service Testing Wizard

☑ Display information message when Web Service Testing Wizard finishes

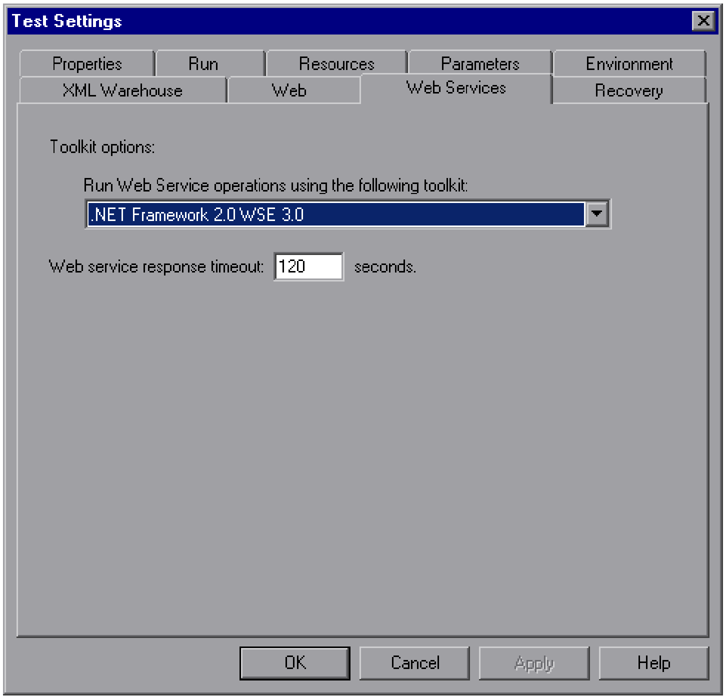
The Web Services tab (**Tools** > **Options**) includes the following options:

Option	Description
Learn new test objects using the following toolkit	Specifies the toolkit you want QuickTest Professional to use when learning new WebService test objects: ➤ **.NET Framework 1.1 WSE 2.0** ➤ **.NET Framework 2.0 WSE 3.0** ➤ **Apache Axis 1.x** **Note:** After you install Microsoft .NET Framework 1.1. you can download and install .NET Framework 1.1 WSE 2.0 SP3 from http://www.microsoft.com/downloads/details.aspx?familyid=8070e1de-22e1-4c78-ab9f-07a7fcf1b6aa&displaylang=en. **Tip:** It is recommended to run Web service tests using the same toolkit with which the test was created. For information on setting run options, see "Defining Web Service Test or Component Settings" on page 76.
WS-I validation tool location	Indicates the path containing the WS-I validation tool. Specify the root path of the **wsi-test-tools** folder (and not the **bin** folder containing the executable file). The **wsi-test-tools** folder must be stored on the computer on which the WSDL is scanned. You can download Interoperability Testing Tools 1.1 for Java and .NET from http://www.ws-i.org. The WS-I validation tool tests Web service interoperability by checking compliance with the WS-I Basic Profile. QuickTest uses the path specified here to perform validation checks in the Validate WSDL dialog box. For more information, see "Checking that Your WSDL Meets WS-I Standards" on page 63.
WS-I validation tool platform	Indicates the Web service client platform type—**.NET** or **Java**.

Option	Description
WS-I report options	Specifies the criteria for the assertion results to be included in the validation report. Test assertions are used by the WS-I validation tool to analyze whether a Web service conforms to WS-I standards. The following options are available: ➤ **Report all assertions.** Reports the results of all assertions. ➤ **Report all assertions but passed assertions.** Reports the results of all assertions except those that have a "passed" result. ➤ **Report only failed assertions.** Reports only the results of assertions that have a "failed" result. **Note:** You can use the WS-I validation tool (accessed from the Validate WSDL dialog box—**Tools** > **Validate WSDL**) to test Web service interoperability. It does this by checking compliance with the WS-I Basic Profile. The WS-I validation tool, **Interoperability Testing Tools 1.1**, is a third-party application and is not provided with QuickTest Professional. The tool can be downloaded from the Web Services Interoperability Organization web site at http://www.ws-i.org and must be installed locally.
Show Welcome screen in Web Service Testing Wizard	Specifies whether to display the Welcome screen when you open the Web Service Testing wizard. You can also control this option by clearing the **Show Welcome screen when running the wizard** check box in the Welcome screen.
Display information message when Web Service Testing Wizard finishes	When you select an operation with input arguments while using the Web Service Testing Wizard, the wizard generates default values for the arguments. If this option is selected, then the wizard displays a message when you click **Finish** to remind you that you must replace the automatically generated argument values with valid values. If you select the **Do not show this message again** check box inside the warning message, then the **Display information message when Web Service Testing Wizard finishes** check box in the Options dialog box is automatically cleared. If you want to reactivate the reminder message, select this check box again.

Defining Web Service Test or Component Settings

The Web Services settings enable you to define the way in which QuickTest runs Web service steps. (You open the Settings dialog box by selecting **File > Settings** in your test or in your component's application area.)



Toolkit options:

Run Web Service operations using the following toolkit: .NET Framework 2.0 WSE 3.0

Web service response timeout: 120 seconds.

Note: The example above shows the Web Services tab of the Test Settings dialog box.

The Web Services tab of the Business Component Settings dialog box displays these settings in read-only format. To modify these settings for components, use the Web Services tab of the Application Area Settings dialog box, which provides the same set of options as shown above.

The Web Services tab includes the following options:

Option	Description
Run Web Service operations using the following toolkit	Enables you to select the toolkit you want QuickTest to use when running Web service operations. For new tests and components, the default toolkit is the same as the toolkit set in the Web Services tab of the Options dialog box for learning Web service objects. **Tip:** Make sure you run Web service tests using the same toolkit with which the test was created. You can verify the toolkit by viewing the **Comment** field in the object repository. (The **Comment** field is located in the **Object Properties** pane.) **Note:** The run toolkit you select is displayed in the Result Details tab of the Test Results window when the top branch is selected in the test results tree.
Web service response timeout	Defines the length of time in seconds that QuickTest attempts to communicate with the Web service during a Web service step. If the time is exceeded, the step fails.

Working with Web Service Operations

A test consists of statements (steps) coded in Microsoft VBScript. These steps are composed of objects, methods, and/or properties that instruct QuickTest to perform operations on, or retrieve information from, your Web service. You can manually insert and edit steps in the Keyword View when working with tests or components, or in the Expert View when working with tests. You can also generate steps automatically using the Web Service Testing Wizard (described on page 45).

You can add steps using the WebService test object, methods, and properties, or the operations of the Web service you are testing.

When you use IntelliSense in the Expert View, or when you select the required operation from the **Operation** column in the Keyword View or the **Operation** box in the Step Generator, the operations are grouped according to the source from which they were derived. The top part of the list contains the operations that are defined in the WSDL. The bottom part of the list contains the WebService message management objects and the operations that are specific to the QuickTest WebService test object.

Using QuickTest WebService Test Object Methods and Properties

The QuickTest Professional Web Services Add-in supplies methods and properties to test the behavior of your Web service including working with message headers and attachments, setting client configurations and proxy information, protecting messages sent from QuickTest by applying security tokens, and more. These QuickTest Web service operations are available in addition to the Web service-specific operations defined in your Web service.

The Web service object model comprises several types of objects: the **WebService** test object, the **WSUtil** object (a utility object), and several **message management objects** (tests only).

Message Management Objects (Expert View)

Message management objects provide methods that enable you to control the way in which QuickTest sends messages to your Web service. A unique message management object exists for each of the following categories: **Attachments**, **Configuration**, **headers**, and **Security**. You return message management objects using the corresponding WebService message management property. For example, the **WebService.Attachments** property returns the **Attachments** object. WebService message management objects are not test objects, and they are not stored in an object repository. Message management objects are available only for tests.

Each message management object supports related message management operations. For example, the **Configuration** object supports configuration-related message management operations, such as SetClientConfiguration and SetProxy. The **Security** message management object supports security-related operations that enable you to add various types of tokens to a step, and so forth.

To fully take advantage of the QuickTest Web service object model, you need to work in the Expert View, as the operations for message management objects are not available in the Keyword View or using the Step Generator.

When inserting steps manually in the Expert View, you can use IntelliSense to add a WebService message management property to a step, and then apply clearly defined operations to that step.

For example, if you add the **Security** WebService message management object, you can select a method for a specific token type, such as SetNetworkCredential, as shown in the following example:

```
WebService("FlightNetWebService").Security.SetNetworkCredential
    "MyUsername","MyPwd","MyDomain"
```

Although the WebService message management objects—**Attachments**, **Configuration**, **Headers**, and **Security**—are available in the Keyword View and the Step Generator, their methods and properties must be initially defined in the Expert View. Therefore, it is recommended not to use these WebService message management objects if you prefer to work in the Keyword View, or if you prefer to insert steps using the Step Generator. Instead, you can associate the **Web_Services.txt** function library with your test and use the alternative operations available from that function library. For information on associating function libraries with your test, refer to the *QuickTest Professional User's Guide*.

WebService Objects (Keyword View)

If you prefer to create or modify Web service-based test steps in the Keyword View, or if you are working with component steps, you can use the functions defined in the **Web_Services.txt** function library provided with the Web Services Add-in to perform similar types of operations. The operations and arguments available in this function library are more generic than the operations and arguments that can be used in the Expert View. For example, instead of using a method, such as AddX509Token, that indicates a specific token type and whose arguments are relevant for this type of token (such as TokenDirection and X509Data), you would use a general method, such as AddSecurityToken. The values you need to enter for this method's arguments depend on the type of security token you want to use.

To use the operations defined in these function libraries, you must first associate the **Web_Services.txt** function library with your test or with your component's application area. For more information, see "Working with Function Libraries" on page 31.

For more information, refer to the **Web Services** section of the *QuickTest Professional Object Model Reference*.

Accessing Operations Supported by the Web Service You Are Testing

In addition to the test object methods supported for every WebService test object, you can use standard statement completion options for operations supported by your Web service. The Web service operations are automatically added to IntelliSense when the test object is created. You can access these operations in the top part of the IntelliSense list displayed for your WebService test object.

You can also select operations supported by your Web service from the list of operations displayed in the **Operation** column in the Keyword View and in the **Operation** box in the Step Generator. The Web service-specific operations are displayed in the top part of the list. The test object methods supported for every WebService test object are displayed in the bottom part.

Working with the .NET Framework Toolkit

If you are creating a Web service test using a .NET Framework toolkit, and you want to perform one or more Web service operations that require parameter data to be sent in a message header, you need to add this information to your test using a **SetHeaderField_<FieldName>Value** method (where **<FieldName>** represents the name of the .NET field in which the parameter data is stored and **Value** is appended to the field name).

When QuickTest learns a WebService test object using the .NET Framework toolkit, it analyzes the Web service's supported operations and automatically creates a **SetHeaderField_<FieldName>Value** method for each operation that requires parameter data (according to the header elements defined in the corresponding WSDL). For example, for the LicenseInfo header element, QuickTest creates the SetHeaderField_LicenseInfoValue method. Note that the *<FieldName>* value is not always identical to the actual header element as it appears in the WSDL.

When you run the WebService Testing Wizard, the Select Service and Operations screen automatically displays these special QuickTest methods in the **Available Operations** list. If you select operations in the Select Service and Operations screen that require parameter data in the header, you should also select the appropriate **SetHeaderField_<FieldName>Value** methods, making sure that the **SetHeaderField_<FieldName>Value** methods are located higher in the **Selected Operations** list than the Web service operations they support.

After a **SetHeaderField_<FieldName>Value** method is called, QuickTest saves the header field value you specify (as if you are working directly with a .NET client). Thereafter, QuickTest inserts this header field value every time it sends a message requiring this header. If a response message updates this header field value, the updated header field value is saved and is used for all subsequent method calls (unless you use the **SetHeaderField_<FieldName>Value** method to modify the header field value again).

You must insert a step with the appropriate **SetHeaderField_<FieldName> Value** method prior to any step containing a Web service operation supported by this method. When you insert a step on a WebService test object that was learned using the .NET Framework toolkit, IntelliSense displays the automatically generated methods in the top part of the operations list, together with the other Web service-specific operations. If you insert the **SetHeaderField_<FieldName>Value** method using the Web Service Testing Wizard, an XML structure parameter is automatically created and populated with default values.

You should edit the XML structure parameter to provide the appropriate data. If you are working in the Keyword View or the Expert View, you must define and populate the XML structure parameter manually. For more information on XML structures, see "Working with XML Structures" on page 91. Note that the **SetHeaderField_<FieldName>Value** method parameter value is always a complex value.

Working with Business Process Testing

When working with Business Process Testing, it is recommended to associate the **Web_Services.txt** function library with your application areas. This function library provides keywords that wrap additional functionality for Web service-based steps, enabling you to perform commonly used Web service operations on any associated component. You can use these operations to perform checks and output values on XML elements, and to perform operations that are similar to the operations available for tests in the Expert View. For more information, see "Working with Function Libraries" on page 31 and refer to the *QuickTest Professional Object Model Reference*.

Guidelines When Working with Business Process Testing

You should consider the following points when working with business components in Business Process Testing.

- If you want to parameterize output values for a component step, use XML structures, or local or component parameters. For more information, see "Parameterizing XML Values" on page 101.
- During a run session, it is recommended to use the same toolkit that QuickTest used to learn your WebService test objects. If you want to work with more than one toolkit, you can create separate components for each toolkit. For more information, see "Setting Web Services Test Options" on page 73 and "Defining Web Service Test or Component Settings" on page 76.
- External resources for components are stored in the associated application area. These include function libraries, object repositories, and recovery scenarios. If you need additional functionality, you or an Automation Engineer can wrap the required functionality in other function libraries and associate them with your application area. It is recommended not to overwrite the existing functionality in any function library supplied with QuickTest, as QuickTest function libraries may be overwritten during an upgrade.

For more information on working with Business Process Testing in QuickTest, refer to the *QuickTest Professional for Business Process Testing User's Guide*. For more information on working with Business Process Testing in Quality Center, refer to the *Mercury Business Process Testing User's Guide*.

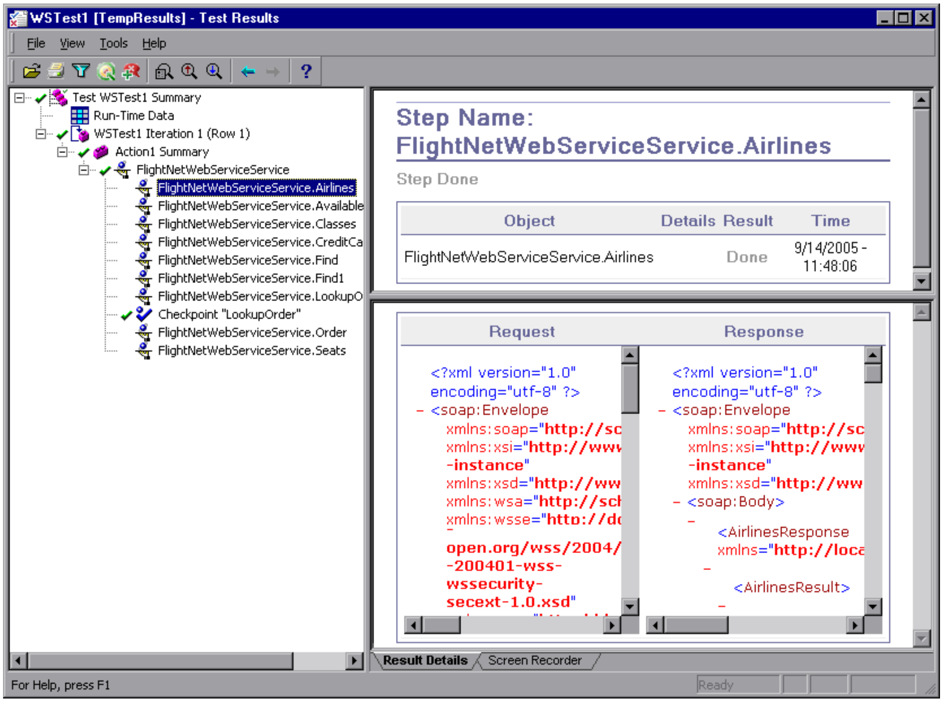
Analyzing the Results of a Web Service Test

You can view the results of a Web service run session in the Test Results window. The window contains a description of the steps performed during the run, as well as a summary of the run results. By default, the Test Results window opens automatically at the end of a run.

Note: You can choose not to open the Test Results window automatically after each run. Choose **Tools** > **Options** and click the **Run** tab. Then clear **View results when run session ends**. You can click the **Results** button or choose **Automation** > **Results** to open the Test Results window when you want to view run results.

If the Web Services Add-in is installed and loaded during a run session, the run toolkit you specified in the Settings dialog box is displayed in the Result Details tab in the Test Results window when the top branch of the tree is selected. The run toolkit is displayed in this tab even if the test or component does not include any Web services object.

Expand the test results tree in the left pane of the Test Results window._



- Test WSTest1 Summary
  - Run-Time Data
  - WSTest1 Iteration 1 (Row 1)
    - Action1 Summary
      - FlightNetWebServiceService
        - FlightNetWebServiceService.Airlines
        - FlightNetWebServiceService.Available
        - FlightNetWebServiceService.Classes
        - FlightNetWebServiceService.CreditCa
        - FlightNetWebServiceService.Find
        - FlightNetWebServiceService.Find1
        - FlightNetWebServiceService.LookupO
        - Checkpoint "LookupOrder"
        - FlightNetWebServiceService.Order
        - FlightNetWebServiceService.Seats

Step Name: FlightNetWebServiceService.Airlines

Step Done

Object	Details	Result	Time
FlightNetWebServiceService.Airlines		Done	9/14/2005 - 11:48:06

Request

```
  <?xml version="1.0"
  encoding="utf-8" ?>
- <soap:Envelope
    xmlns:soap="http://sc
    xmlns:xsi="http://www
    -instance"
    xmlns:xsd="http://ww
    xmlns:wsa="http://sch
    xmlns:wsse="http://dc
    open.org/wss/2004/
    -200401-wss-
    wssecurity-
    secext-1.0.xsd"
```

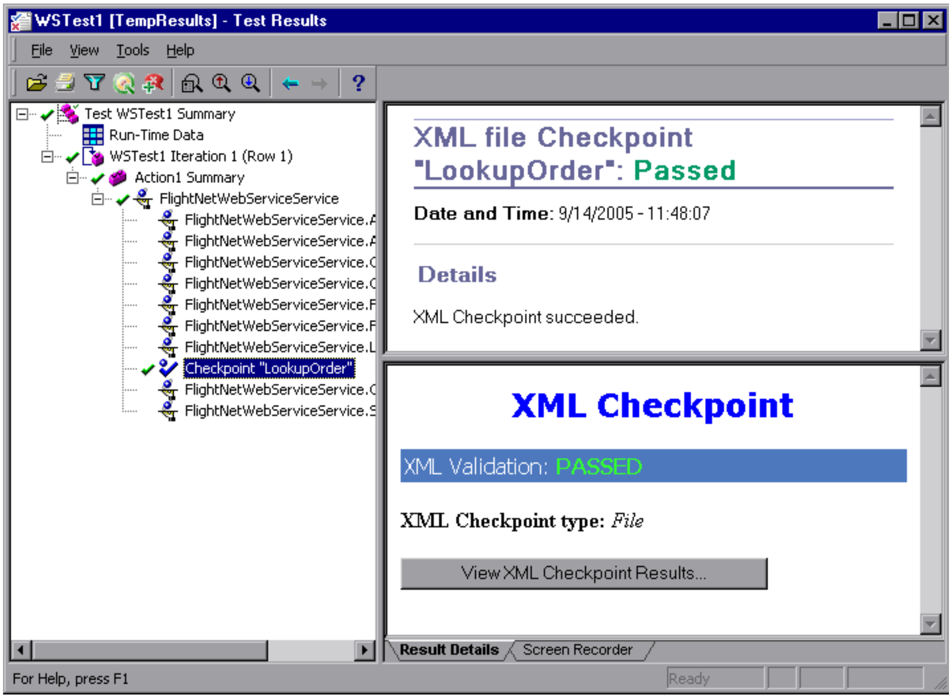
Response

```
  <?xml version="1.0"
  encoding="utf-8" ?>
- <soap:Envelope
    xmlns:soap="http://sc
    xmlns:xsi="http://www
    -instance"
    xmlns:xsd="http://ww
  - <soap:Body>
    -
        <AirlinesResponse
        xmlns="http://loca
      -
          <AirlinesResult>
        -
```

Note: By default, QuickTest captures the detailed requests and responses for all steps.

The Web Services Add-in uses the **Save step screen capture to results** option in the Options dialog box to determine whether to capture requests and responses for your steps. If you do not want QuickTest to capture requests and responses for your steps, choose **Tools** > **Options** and click the **Run** tab. Then choose **Never** from the **Save step screen capture to results** list.

Select a step or checkpoint in the test results tree to view the results for that step in the right-hand pane.



Test WSTest1 Summary
Run-Time Data
WSTest1 Iteration 1 (Row 1)
Action1 Summary
FlightNetWebServiceService
FlightNetWebServiceService.A
FlightNetWebServiceService.A
FlightNetWebServiceService.C
FlightNetWebServiceService.C
FlightNetWebServiceService.F
FlightNetWebServiceService.F
FlightNetWebServiceService.L
Checkpoint "LookupOrder"
FlightNetWebServiceService.C
FlightNetWebServiceService.S

XML file Checkpoint "LookupOrder": Passed

Date and Time: 9/14/2005 - 11:48:07

Details

XML Checkpoint succeeded.

XML Checkpoint

XML Validation: PASSED

XML Checkpoint type: *File*

For checkpoints, the top right pane displays the checkpoint step results. The bottom right pane shows the details of the schema validation (if applicable) and a summary of the step or checkpoint results. If the schema validation failed, the reasons for the failure are also shown.

The bottom right pane may include the **View XML Checkpoint Results** button, enabling you to view details of the checkpoint's failure. The Web Services Add-in uses the **Save step screen capture to results** option in the Run tab of the Options dialog box to determine when to display this button.

For more information about the Test Results window and analyzing test and XML checkpoint results, refer to the *QuickTest Professional User's Guide*.

4

Working with XML Data

This chapter describes the various ways in which you can use QuickTest Professional to manipulate XML data when testing Web services.

About Working with XML Data

Web service operations often return XML data when the operation is performed.

When working with tests, you can create checkpoints and output values on the XML data that is returned from an operation performed on a Web service. You can perform **XMLData** methods on the returned XML data. You can also create XML structures and use them as templates for data in your test. This enables you to parameterize specific values in an XML document and use the parameterized XML hierarchy, for example, as the value for an XML operation argument or as the expected XML element value in a checkpoint.

When working with components, checkpoints and output values are not available. Instead, you can use the functions (keywords) defined in the **Web_Services.txt** function library to perform similar types of steps. To use these functions, the **Web_Services.txt** function library must be associated with the component's application area. For more information, see "Working with Function Libraries" on page 31. If needed, an Automation Engineer can also wrap additional types of functionality in other function libraries and associate them with the component's application area. For more information on working with function libraries and application areas, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

QuickTest provides a set of XMLData methods that you can use to manipulate the XML values returned from the Web service operations in your test or component. For more information, refer to the *QuickTest Professional Object Model Reference*.

Checking XML

When working with tests, you can add XML checkpoints on the returned values from Web service operations in your test. An XML checkpoint is a verification point that compares actual values for specified XML elements, attributes and/or values with their expected values. If the results do not match, the checkpoint fails.

For example, suppose a flight reservation Web service has an operation that sets a flight itinerary and returns XML containing the specified itinerary details. A step in the test of the Web service could select a specific airline and return XML containing flight details. An XML checkpoint could compare the date and time information contained in the flight details XML data with the expected values for those XML nodes and check that the itinerary details match the expected result.

You insert a checkpoint using **Insert** > **Checkpoint** > **XML Checkpoint (From Resource)** or by using the **Insert Checkpoint or Output Value** button. You then select a test object and the operation whose return values you want to check.

Note: You can also insert checkpoints for every step automatically when using the Web Service Testing Wizard. For more information, see "Understanding the Web Service Testing Wizard" on page 45.

After you insert the XML checkpoint, QuickTest creates an expected XML hierarchy based on the return type defined in the WSDL. If the operation return type is not clearly defined, then QuickTest generates a generic XML hierarchy containing two nodes. In either case, QuickTest cannot generate the expected values of the nodes at the time that you insert the checkpoint. To check the relevant information, you must first populate the XML checkpoint hierarchy with the elements, attributes, and values expected in the returned XML. You can populate the XML hierarchy and values using the **Update Run Mode** option, by updating the hierarchy manually, or by importing a hierarchy from an existing file.

For more information about working with XML checkpoints and updating the XML hierarchy for XML checkpoints, refer to the *QuickTest Professional User's Guide*.

After you run your test, you can view summary results of the XML checkpoint in the Test Results window. You can also view detailed results by opening the XML Checkpoint Results window. For more information, see "Analyzing the Results of a Web Service Test" on page 83, and refer to the *QuickTest Professional User's Guide*.

Checking Business Components

When working with components, you can check that the returned values from Web service operations in your test match the expected values using a function library operation, such as **VerifyXMLValue**.

This and other verification operations are available only if the **Web_Services.txt** function library file is associated with your component's application area. For more information on using operations from a function library, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

Outputting XML Values

You can insert XML output value steps to retrieve data from the XML returned from Web service operations. An XML output value is a value captured from an XML return value during the run session and stored in a parameter for use at another point in the run session. When you create an output value step, the test retrieves a value at a specified point during the run session and stores it in the specified parameter. When you want to use the value later in the run session as input, you instruct QuickTest to retrieve it from the parameter.

For example, a step in a test of a flight reservation system could require the retrieval of an order number. This order number could be a parameterized output value from an XML structure, and could be stored in an action parameter in a test. You could then retrieve the order number value from the parameter for use as an operation argument in a later step.

You insert an output value step using **Insert** > **Output Value**> **XML Output Value (From Resource)** or by using the **Insert Checkpoint or Output Value** button.

QuickTest creates an expected XML hierarchy based on the return type defined in the WSDL. If the operation return type is not clearly defined, then QuickTest generates a generic XML hierarchy containing two nodes. To select the nodes you want to output, you must first ensure that the XML output value has the hierarchy you want. You can populate the XML hierarchy by updating the structure manually, importing it from an existing file, or updating it using the Update Run mode. For more information about updating the XML hierarchy for Web service XML output values, refer to the *QuickTest Professional User's Guide*.

When you run your test, you can view summary results of the XML output value step in the Test Results window. You can also view detailed results by opening the XML Output Value Results window.

For more information, refer to the *QuickTest Professional User's Guide*.

Outputting XML Values for Business Components

When working with components, you can insert a step to retrieve a value from an XML object using the **OutputXMLValue** operation.

This operation is available only if the **Web_Services.txt** function library file is associated with your component's application area. For more information on using operations from a function library, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

Working with XML Structures

When you need to supply a complex value of type XML, you can either store the value in an XMLData object or you can use a QuickTest **XML structure**, which provides a visual, editable interface for the hierarchy of the XMLData object. XML structures are available for both tests and components.

XML structures also enable you to parameterize individual node values within the XML hierarchy so that those values can be retrieved from another parameter during the run session.

For example, you may have a Web service that manages an address book. The Web service has an **AddAddr** operation that adds a new entry to the address book. The operation receives an XML data argument containing the name, phone numbers, and address of the new entry.

Suppose you want to run a test that performs the **AddAddr** operation five times to add five different people to the address book, and then retrieves the updated address book. Instead of entering five **AddAddr** statements, each with a different XML string for the argument, you can create one XML structure and parameterize its name, phone number, and address values to take values from the data table (for tests) or from a local parameter (for components). You can then insert one **AddAddr** statement and set its argument to use the XML structure you created. Then, you can run multiple iterations of the action or component containing that step.

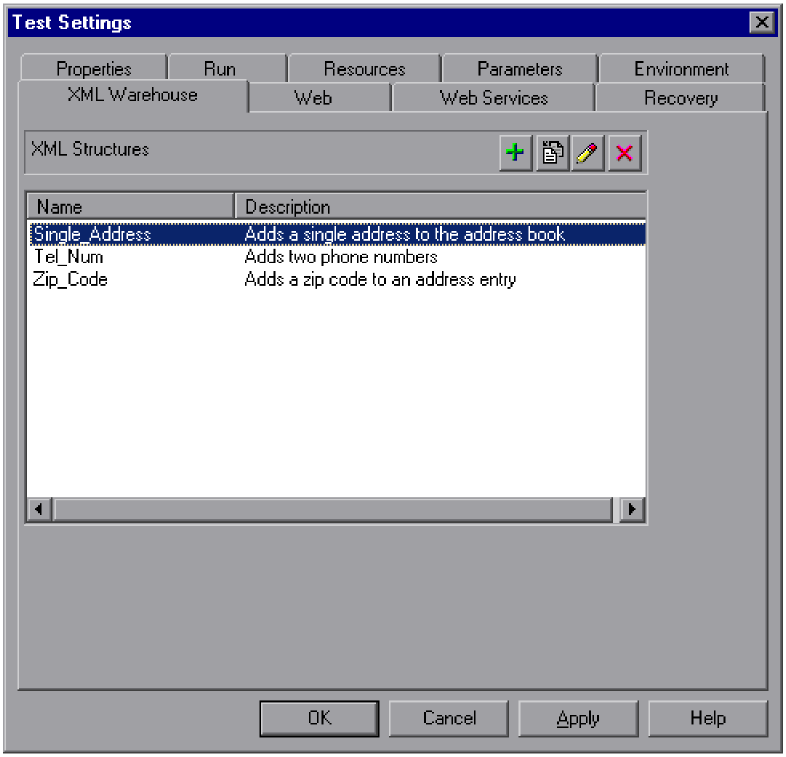
You can create several different XML structures. You can use an XML file or an existing XML structure as the base for a new XML structure, or you can edit nodes manually.

You save XML structures in the XML Warehouse for your test (**File** > **Settings** > **XML Warehouse**). XML structures are stored with the test or component—they are not external resources.

After you create an XML structure, you can parameterize XML data arguments using your XML structures. You can do this in the Keyword View or by manually inserting statements using the Expert View (tests only). For more information, see "Parameterizing XML Values" on page 101, or "Working with XML Data Operations" on page 103.

Managing XML Structures

The XML Warehouse tab displays information about the XML structures that you defined for your test or component. It also enables you to create new XML structures, modify or duplicate existing structures, and remove structures that you no longer require.



Name	Description
Single_Address	Adds a single address to the address book
Tel_Num	Adds two phone numbers
Zip_Code	Adds a zip code to an address entry

The XML Warehouse tab includes the following buttons:

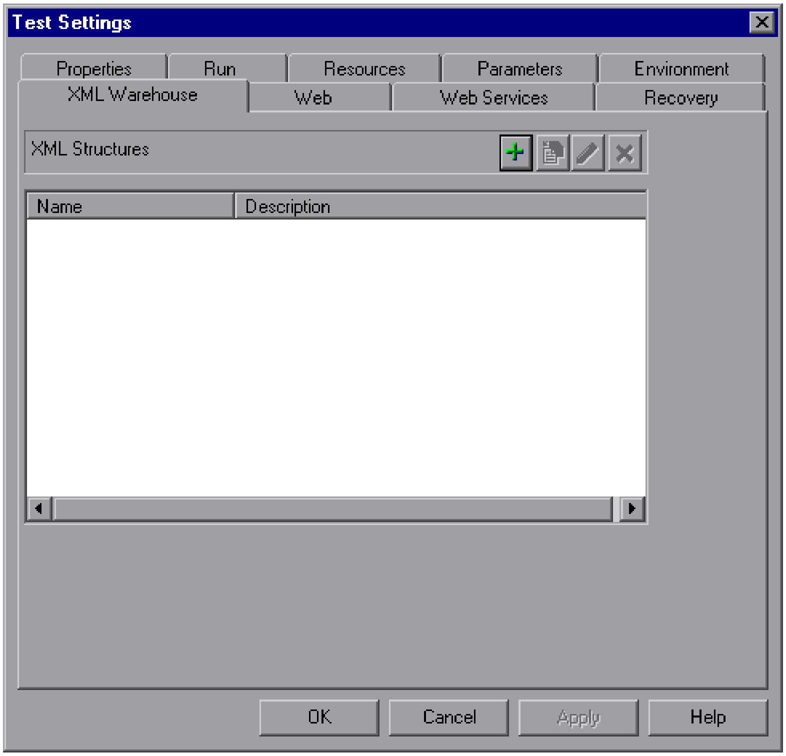
Button	**Description**
+	Enables you to create an XML structure. For more information, see “Creating XML Structures” on page 93.
⎘	Enables you to duplicate an XML structure. For more information, see “Duplicating XML Structures” on page 95.
✎	Enables you to edit an XML structure. For more information, see “Modifying XML Structures” on page 95.
✕	Enables you to delete an XML structure. For more information, see “Deleting XML Structures” on page 96.

Creating XML Structures

You can create XML structures for use as an XML value in your test. You first set up the XML hierarchy for the XML structure by importing it and/or manually adding and editing its nodes. You can then edit or parameterize the attributes and values of the XML structure.

To create an XML structure:

1 Click the **Settings** toolbar button or choose **File** > **Settings**. The Test Settings or Business Component Settings dialog box opens. Click the **XML Warehouse** tab.



Note: The XML Warehouse tab of the Business Component Settings dialog box is identical to the example shown above.

---

2 Click the **Create new XML structure** button. The Create New XML Structure dialog box opens.

3 Enter a name and description for your XML structure.

4 Create the new XML structure, as described in “Understanding the Create New/Edit/Duplicate XML Structure Dialog Box” on page 96.

5 Click **OK** to create the XML structure and save it in the XML Warehouse for your test.

Duplicating XML Structures

If you already created an XML structure, you can create a new XML structure based on an existing one. This is useful if you need a new XML structure with the same hierarchy as an existing one, but you want to use different values or parameterization settings.

To duplicate an XML structure:

1 In the XML Warehouse tab of the Test Settings or Business Component Settings dialog box, select the XML structure you want to duplicate.

2 Click the **Duplicate XML Structure** button. The Duplicate XML Structure dialog box opens.

3 Specify a name for the new XML structure.

4 Modify the XML name, description, hierarchy elements and/or parameterization options, as described in “Understanding the Create New/Edit/Duplicate XML Structure Dialog Box” on page 96.

5 Click **OK**. The new XML structure is added to the list in the XML Warehouse tab.

Modifying XML Structures

If you already created an XML structure, you can modify its settings. For example, you can define or modify parameterization of values or attributes, or add or delete specific elements in the XML structure.

To modify an XML structure:

1 In the XML Warehouse tab of the Test Settings or Business Component Settings dialog box, select the XML structure you want to modify.

2 Click the **Edit XML Structure** button. The Edit XML Structure dialog box opens.

3. Modify the hierarchy elements and/or parameterization options, as described in “Understanding the Create New/Edit/Duplicate XML Structure Dialog Box” on page 96.
4. Click **OK** to save your changes and close the Edit XML Structure dialog box.

Deleting XML Structures

You can delete XML structures from the XML Warehouse if you no longer need them for your test or component.

To delete an XML structure:

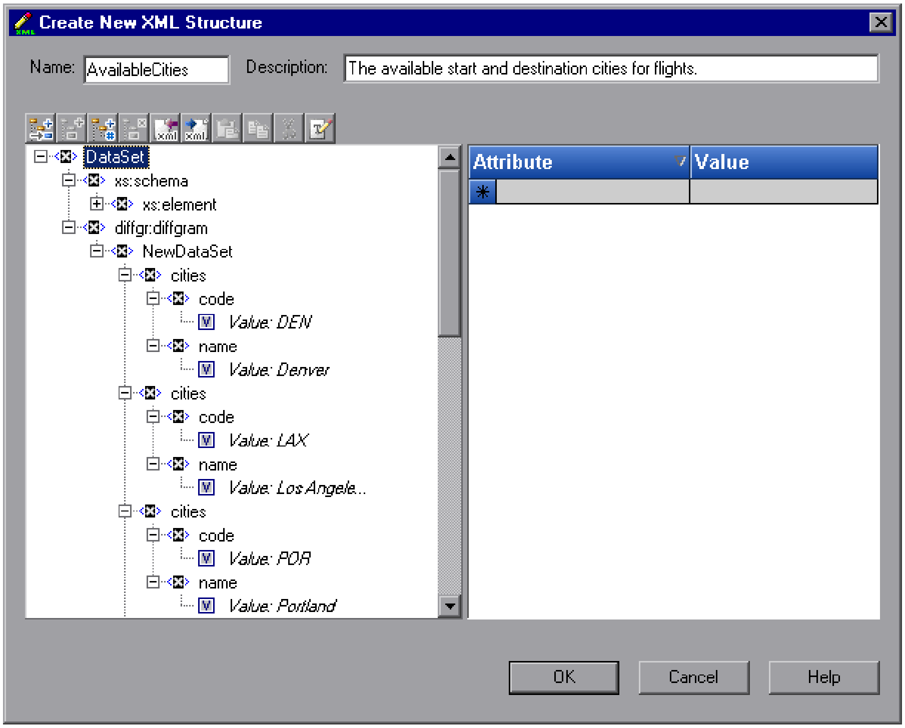
1. In the XML Warehouse tab of the Test Settings dialog box, select the XML structure you want to delete.
2. Click the **Delete XML Structure** button. A confirmation message opens.
3. Click **Yes** to delete the selected XML structure from the XML Warehouse.

Understanding the Create New/Edit/Duplicate XML Structure Dialog Box

Note: Although this section refers specifically to the Create New XML Structure dialog box, the information provided also applies to the Edit XML Structure dialog box and the Duplicate XML Structure dialog box.

The Create New/Edit/Duplicate XML Structure dialog box enables you to create and modify an XML structure and save it in the XML Warehouse.

The dialog box displays the element hierarchy, attributes, and values (character data) of the XML hierarchy you create.



Create New XML Structure

Name: AvailableCities

Description: The available start and destination cities for flights.

```
DataSet
  xs:schema
    xs:element
  diffgr:diffgram
    NewDataSet
      cities
        code
          Value: DEN
        name
          Value: Denver
      cities
        code
          Value: LAX
        name
          Value: Los Angele...
      cities
        code
          Value: POR
        name
          Value: Portland
```

XML Structure Information

The top part of the Create New/Edit/Duplicate XML Structure dialog box displays information about the XML structure:

Item	Description
Name	The XML structure name. When you are editing an existing structure, this box is read-only.
Description	A textual description of the XML structure.

In addition, the following commands are available according to the node you select in the XML tree:

Command	Icon	Description
Add Child	+	Adds a child node below the selected node in the XML tree.
Insert Sibling	+	Adds a sibling node at the same level as the selected node in the XML tree.
Add Value	#	Enables you to assign a constant or parameterized value to the selected item.
Delete	×	Deletes the selected node.
Import XML	xml	Enables you to browse to and select another XML file. The new file will override the selected node's current sub-tree.
Export XML	xml	Enables you to save the current XML file.
Paste	📋	Pastes a cut or copied node as a child node below the selected node in the XML tree. **Note:** You cannot paste an XML element node as its own descendant.
Copy	⧉	Makes a copy of the selected node, which you can then paste in another location in the XML tree.
Cut	✂	Prepares the selected node to be cut and copies it to the clipboard. When you paste the node in the new location, it is removed from the original location in the XML tree.

Command	**Icon**	**Description**
Edit XML as Text	✎	Opens the Edit XML as Text dialog box, enabling you to modify the XML text of the selected node and its subnodes in a text editor. This dialog box is used mainly for constructing an entire XML segment from a string or for fixing syntax problems that prevent the dialog box from displaying the XML tree correctly. It is also useful when you want to use copy-paste functionality to edit the XML tree. For more information, refer to the *QuickTest Professional User's Guide*.
Duplicate		Adds a new node, identical to the selected one, as a sibling node at the same level as the selected node in the XML tree. **Note:** This command is available only from the context menu (Right-click menu).

Choosing the Element Values and/or Attributes to Edit or Parameterize

The Create New/Edit/Duplicate XML Structure dialog box contains an XML tree that displays the hierarchy of the XML structure, enabling you to select the element values and/or attributes that you want to edit or parameterize. You can edit or parameterize only the values; you cannot edit or parameterize element tags. This pane contains the following areas:

Area	Description
XML Tree	The XML tree displays the hierarchal relationship between each element and value in the XML file. Each element is displayed with a <▣> icon. Each value is displayed with a V icon. **Note:** Expanding a large XML tree may take some time.
Data Area	When an element is selected in the XML tree, the data area to the right of the tree displays the element's attributes and values. When an element value is selected in the XML tree, the data area displays the value node's data. When you click in a **Value** cell, the parameterization button <#> is displayed. Click the button to parameterize the value. For more information, see "Parameterizing XML Values" on page 101.

Tips:

- The XML tree pane and the **Attribute** and **Value** columns in the right pane are resizable.
- You can delete an attribute by selecting its row and pressing the DELETE key on your keyboard.

Parameterizing XML Values

When you add a step to your test, you can parameterize operation arguments of type XML so that they use dynamic data from an XML structure that you saved in the XML Warehouse. When you select to parameterize an XML argument, the Value Configuration Options dialog box opens. You can create a new XML structure from within this dialog box, and then use this XML structure as the parameterized value.

To parameterize an XML operation argument:

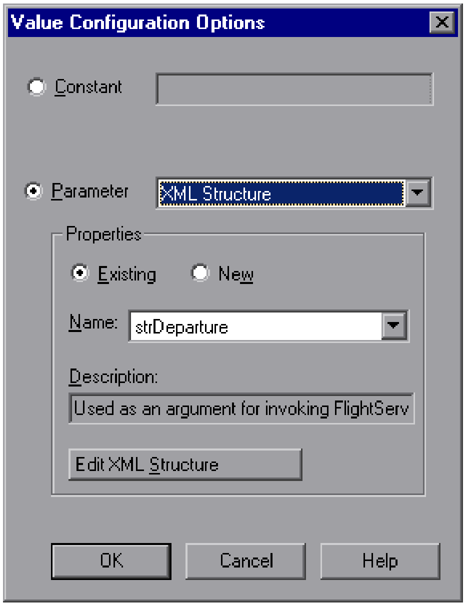
1 In the Keyword View, click in the **Value** column of the operation that you want to parameterize.

Note: The **Value** cell can comprise more than one partition, containing different argument values for the selected operation. Clicking a partition displays the parameterization button <#>, as well as information for each argument in a tooltip. In the tooltip, the argument for the partition that is currently highlighted is displayed in bold.

2 Click the parameterization button <#>. The Value Configuration Options dialog box opens, showing the currently defined value.

3 Select **Parameter**. If the value is already parameterized, the **Parameter** section displays the current parameter definition for the value. If the value is not yet parameterized, the **Parameter** section displays the default parameter definition for the value.

4 From the **Parameter** list, select **XML Structure**.



Value Configuration Options

○ Constant

● Parameter: XML Structure

Properties

● Existing ○ New

Name: strDeparture

Description: Used as an argument for invoking FlightServ

5 In the **Properties** area, select the type of XML structure you want to use:

- ➤ **Existing.** Enables you to specify an XML structure you have already created.

  If you select **Existing**, select the name from the **Name** list. If you want to modify the existing structure, click the **Edit XML Structure** button and go to step 6. Otherwise, click **OK**. The test uses the values from the XML structure you specify. If you specify an existing XML structure, the description is displayed in read-only mode.

- ➤ **New.** Enables you to create a new XML structure.

  If you select **New**, enter a name and textual description for it in the **Name** and **Description** boxes and click **Create New XML Structure**.

The Create New/Edit XML Structure dialog box opens.

6 Create or edit the XML structure as described in “Understanding the Create New/Edit/Duplicate XML Structure Dialog Box” on page 96.

7 When you finish, click **OK** to close the Create New/Edit XML Structure dialog box.

8 Click **OK** to close the Value Configuration Options dialog box.

Working with XML Data Operations

A test consists of statements (steps) coded in Microsoft VBScript. These statements are composed of objects, methods, and/or properties that instruct QuickTest to perform operations on, or retrieve information from, your Web service. You can insert and edit statements in your test manually in the Keyword View (tests and components) or in the Expert View (tests only).

You can add steps to perform XML data operations using the XMLWarehouse objects (tests and components) and XMLData objects (tests only).

Specifying XML Structures for XML Value Types

The XMLWarehouse object enables you to parameterize XML values such as method arguments of type XML. This enables you to use data from an XML structure that you saved in the XML Warehouse. The individual values of the elements and attributes in the XMLWarehouse object can be parameterized so that dynamic data can be used during a run session.

The following example uses the XMLWarehouse object in the Expert View to use the parameterized **Departure** XML structure as the argument for the **Find** method.

```
Set Find=
WebService("FlightNetWebService").Find(XMLWarehouse("Departure"))
```

When inserting steps manually in the Expert View, you can use IntelliSense to add an XML structure from the XML Warehouse. When you type XMLWarehouse followed by an open parenthesis **(**, QuickTest displays a list of all XML structures in the XML Warehouse.

After inserting an XMLWarehouse object in a step in the Expert View, you can right-click on the object and select **View/Edit XML Structure** to open the Edit XML Structure dialog box.

Working with Retrieved XML Data

Many Web service operations return XML data objects. You can check and/or manipulate pieces of these objects using XMLData and other XML-related methods.

For example, you can use the **XMLData.ChildElementsByPath** method to return the child elements of a returned XML data object as an XMLElementsColl collection object. You can then use the **XMLElementsColl.Count** property to check whether the collection has the expected number of elements.

For information about the XMLData object, refer to the **Supplemental** section of the *QuickTest Professional Object Model Reference*.

Index