



MERCURY BUSINESS PROCESS TESTING™

VERSION 9.0

Tutorial

MERCURY™

Mercury Business Process Testing

Tutorial
Version 9.0

Mercury Business Process Testing Tutorial, Version 9.0

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332; 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

© 2006 Mercury Interactive Corporation, All rights reserved

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

Table of Contents

| | |
|---|-----------|
| Welcome to the Business Process Testing Tutorial | v |
| Before You Begin | vi |
| Using This Tutorial..... | vii |
| Typographical Conventions..... | ix |
| Lesson 1: Introducing Business Process Testing | 1 |
| About Business Process Testing | 2 |
| Understanding Business Process Testing Roles and Workflow | 2 |
| Starting Business Process Testing | 6 |
| Exploring the Business Components Module Window..... | 8 |
| Exploring the Test Plan Module Window..... | 11 |
| Exploring the Mercury Flight Reservations Application..... | 14 |
| Lesson 2: Creating Business Components | 17 |
| Creating Components | 18 |
| Defining Component Details | 20 |
| Adding Manual Steps..... | 22 |
| Lesson 3: Creating Business Process Tests | 25 |
| Creating a Business Process Test | 25 |
| Adding Components to a Business Process Test | 27 |
| Lesson 4: Automating Components..... | 33 |
| Converting Components to Automated Components | 34 |
| Selecting Application Areas | 36 |
| Lesson 5: Adding Keyword-Driven Component Steps | 39 |
| Creating Automated Steps for the Login Component..... | 40 |
| Creating Automated Steps for the Create Order Component | 47 |
| Creating Automated Steps for the Update Order Component | 53 |
| Creating Automated Steps for the Logout Component..... | 57 |

| | |
|--|------------|
| Lesson 6: Debugging Business Process Tests..... | 61 |
| Debugging Your Business Process Test..... | 62 |
| Lesson 7: Working with Parameters | 67 |
| Defining an Output Component Parameter for the Create Order Component..... | 68 |
| Defining Input Parameters for the Create Order and Update Order Components..... | 70 |
| Creating the Delete Order Component and Adding It to Your Test..... | 74 |
| Iterating Components..... | 77 |
| Debugging the Parameterized Test..... | 83 |
| Grouping Components | 89 |
| Lesson 8: Running Business Process Tests | 93 |
| Selecting Iterations to Run | 94 |
| Creating a Run-Time Parameter | 97 |
| Setting Up a Business Process Test in the Test Lab Module..... | 100 |
| Running the Business Process Test in the Test Lab Module | 104 |
| Viewing Run Results..... | 107 |
| Lesson 9: Where Do You Go from Here? | 113 |
| Getting Started | 113 |
| Getting Additional Information | 114 |
| Documentation Updates | 115 |

Welcome to the Business Process Testing Tutorial

Welcome to the Business Process Testing tutorial.

Business Process Testing with Quality Center enables non-technical Subject Matter Experts to build and work with business components in a script-free environment. A **business process test** is a scenario comprising a serial flow of business components, designed to test a specific business process of an application.

This tutorial is a self-paced guide that teaches you the basics of Business Process Testing. You access Business Process Testing from Quality Center. The tutorial will familiarize you with the process of creating, implementing, and running keyword-driven business components in business process tests.

This tutorial focuses on creating manual component steps, and then using them as a basis for creating automated steps in keyword-driven components. Manual steps can also be used to run both manual and automated business components manually, using the Manual Runner. For more information, refer to the *Mercury Business Process Testing User's Guide*.

This introduction describes:

- Before You Begin
- Using This Tutorial
- Typographical Conventions

Before You Begin

This tutorial teaches you how to create business process tests using keyword-driven components.

This tutorial does not describe working with WinRunner components. For more information on WinRunner components, refer to the *Mercury Business Process Testing User's Guide* and the *Mercury WinRunner User's Guide*.

To do this tutorial, you must have access to:

- ✓ **Quality Center 9.0**—Mercury's Web-based test management tool. You must have licenses available for Quality Center and Business Process Testing and be able to access these modules: **Business Components**, **Test Plan**, and **Test Lab**.
- ✓ **QuickTest Professional 9.0**—the testing tool used to run the automated business process tests used in this tutorial. You must:
 - Have the proper version of QuickTest Professional installed on your computer.
 - Enable the **Allow other Mercury products to run tests and components** option in QuickTest (**Tools > Options > Run** tab) to work with Quality Center. For more information, refer to the *Mercury QuickTest Professional User's Guide*.
 - Install the QuickTest Professional Add-in and the Mercury Quality Center Connectivity Add-in from the Mercury Quality Center Add-ins page. For more information, refer to the *Mercury Quality Center Installation Guide*.

- ✓ **QualityCenter_Demo**—a demonstration project that helps introduce you to Quality Center.

Note: If the Quality Center demo project is not installed on your server, or if the project has been modified by other users (if business components or business process tests have already been defined in the project), contact your Quality Center administrator to get a new (clean) copy of the demo project installed on your server.

- ✓ **Mercury Flight Reservations**—a sample application that simulates an application for reserving flights. This application is automatically installed with QuickTest Professional.

Using This Tutorial

The tutorial is divided into nine lessons, in which you will create and run business components on the sample Mercury Flight Reservations application. After completing the tutorial, you can apply the skills you have learned to your own application.

Lesson 1, Introducing Business Process Testing introduces you to Business Process Testing, provides an overview of the Business Process Testing user interface, and familiarizes you with the sample Mercury Flight Reservations application.

Lesson 2, Creating Business Components teaches you how to create non-automated components and define component details.

Lesson 3, Creating Business Process Tests teaches you how to build business process tests by creating a serial flow of business components.

Lesson 4, Automating Components teaches you how to convert your non-automated components to keyword-driven components and to attach an application area to each of your business components.

Lesson 5, Adding Keyword-Driven Component Steps teaches you how to add keyword-driven steps to your business components.

Lesson 6, Debugging Business Process Tests teaches you how to run a business process test and identify problems.

Lesson 7, Working with Parameters teaches you how to define input and output component parameters.

Lesson 8, Running Business Process Tests teaches you how to run business process tests and view the results.

Lesson 9, Where Do You Go from Here? explains how to start testing your own applications and where to find more information on Business Process Testing.

Note: These lessons follow the workflow described in Lesson 1, “Introducing Business Process Testing.”

Typographical Conventions

This guide uses the following typographical conventions:

| | |
|------------------------------|--|
| UI Elements | 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.” |
| <i>Arguments</i> | This style indicates method, property, or function arguments and book titles. For example, “Refer to the <i>Mercury User’s Guide</i> .” |
| <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.” |
| Function_Name | This style indicates method or function names. For example, “The wait_window statement has the following parameters:” |
| [] | 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. |

Welcome

1

Introducing Business Process Testing

This tutorial provides a step-by-step guide to creating and running keyword-driven business components and business process tests, using the Business Components, Test Plan, and Test Lab modules in Quality Center.

This tutorial assumes that you are familiar with Quality Center procedures or have at least completed the Quality Center tutorial. For more information, refer to the *Mercury Quality Center User's Guide* and the *Mercury Quality Center Tutorial*.

In this lesson you will learn about the following:

- About Business Process Testing
- Understanding Business Process Testing Roles and Workflow
- Starting Business Process Testing
- Exploring the Business Components Module Window
- Exploring the Test Plan Module Window
- Exploring the Mercury Flight Reservations Application

About Business Process Testing

Business Process Testing enables Subject Matter Experts to design quality assurance tests early in the development cycle and in a script-free environment.

Business Process Testing is based on the creation, implementation, and running of business components in business process tests using the Quality Center **Business Components**, **Test Plan**, and **Test Lab** modules.

- The Business Components module enables you to create, manage, and automate reusable business components, comprising steps that perform specific tasks in a business process.
- The Test Plan module enables you to drag and drop the components into business process tests, and debug the components.
- The Test Lab module enables you to run business process tests and view the results.

Understanding Business Process Testing Roles and Workflow

The Business Process Testing model is role-based, allowing non-technical Subject Matter Experts to work on tests with Automation Engineers, both together and in parallel. The Business Process Testing model also follows a workflow that is supported by the user interface design. By following the roles and workflow that Business Process Testing offers, you can maximize the quality and efficiency of the testing process in your organization.

Identifying Business Process Testing Roles

The following two basic user roles are identified in the Business Process Testing model:

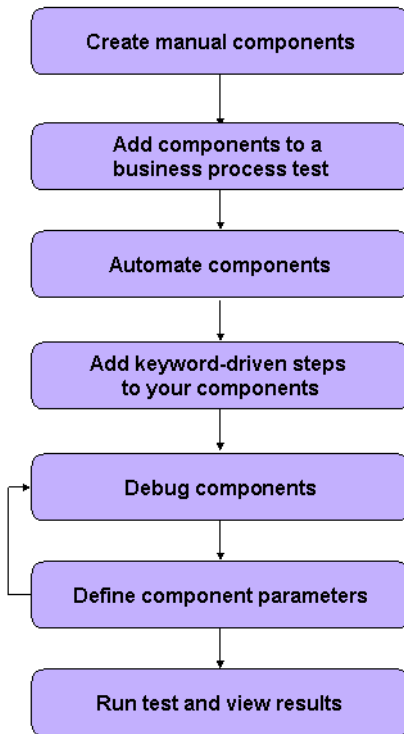
- **Automation Engineer**—Creates application areas and the object repository, library files, and recovery scenarios associated with these application areas using a testing tool such as QuickTest Professional or WinRunner. The Automation Engineer can also assist with the debugging process after the Subject Matter Expert creates automated business process tests.

- **Subject Matter Expert**—Creates and automates components in the Business Components module of Quality Center, combines the business components into business process tests, composed of a serial flow of the components, and uses the resources defined by the Automation Engineer to add object and function steps using the keyword view. The Subject Matter Expert can check the components within business process tests by running and debugging them from Test Plan and can then include them in test set runs to test the application.

For the purposes of this tutorial, the Quality Center project provides the application area, object repository and library file resources that an Automation Engineer would normally provide, so that you can focus on learning the Subject Matter Expert role.

Note: These roles are flexible, and depend on the abilities and time resources of the personnel using Business Process Testing. There are no limitations or product-specific rules controlling which roles must be defined in a particular organization, or which types of users can do which Business Process Testing tasks (provided that the users have the correct permissions).

The Business Process Testing workflow in this tutorial consists of the following main phases:



1 Create manual business components

Create manual business components in the Business Components module by defining details and manual steps.

2 Add components to a business process test

Drag and drop your business components into a business process test.

3 Automate components

Convert each business component to an automated keyword-driven component and select an application area to use with each component.

4 Add keyword-driven steps to your components

Define the automated steps that perform specific tasks in each business component.

5 Debug components

Run your business components in the test to ensure that they perform successfully and without interruption. This phase should be repeated after each significant enhancement to the test.

6 Define component parameters

Expand the scope of your business components and business process tests by replacing fixed values with component parameters.

7 Run test and view results

Run your business process test and view the results to pinpoint defects in your application.

Before you begin creating business components, you should familiarize yourself with the module windows used in Business Process Testing, and the Mercury Flight Reservations Sample Application. These topics are described in the next sections.

Starting Business Process Testing

You start working with Business Process Testing by logging in to Quality Center and opening the Business Components module.

To start working with Business Process Testing:

1 Open the Mercury Quality Center Options window.

In your Web browser, enter your Mercury Quality Center URL:
`http://<Mercury Quality Center server name>/<virtual directory name>`

The Mercury Quality Center Options window opens.



2 Open Quality Center.

Click the **Quality Center** link.

The first time you run Quality Center, the application is downloaded to your computer. Then, each time you open Quality Center, it automatically carries out a version check. If Quality Center detects a newer version, it downloads the latest version to your computer.

The Mercury Quality Center Login window opens.

Quality Center

User Name:

Password:

☐ Automatically log in to my last domain and project on this machine

Authenticate

Domain:

Project:

Login

Optimize Application Quality

3 Log in to your Business Process Testing tutorial project.

In the **User Name** box, enter the following user name: **alex_qc**.

Leave the **Password** box blank. The above user names do not have passwords assigned to them.

Click the **Authenticate** button. The **Domain** and **Project** boxes are enabled.

In the **Domain** list, select the **DEFAULT** domain.

In the **Project** list, select the **QualityCenter_Demo** project.

Click the **Login** button.

The Welcome to Mercury Quality Center window may open. Select **Don't show this again** and click **Close**.

The Quality Center main window opens. The domain name, project name and user name you entered are displayed in the title bar.

Note: After login, Quality Center displays the module in which you (the user name you entered) last worked.



If the Business Components module is not currently displayed, click the **Business Components** module button in the sidebar. The Business Components module opens.

Exploring the Business Components Module Window

Business components are created in the Quality Center Business Components module. In this section you will explore the main elements of this module.

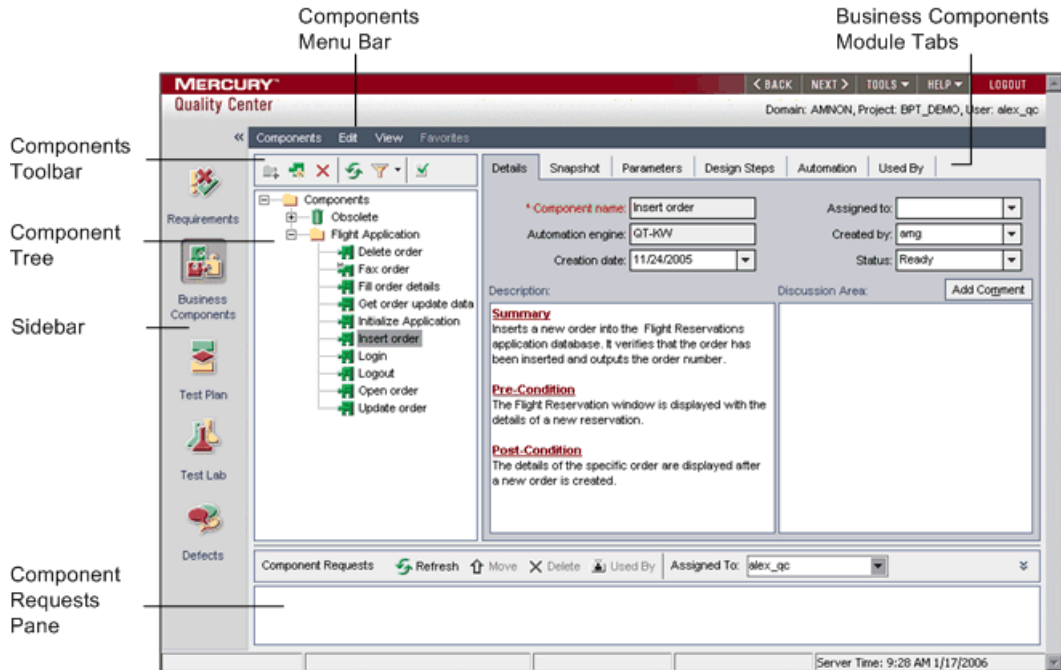
1 Open the Business Components module.



If the Business Components module is not currently displayed in Quality Center, click the **Business Components** module button in the sidebar. The Business Components module opens.

2 Explore the elements of the Business Components module.

An example of the Business Components module window is shown below, displaying the Details tab for a keyword-driven component.

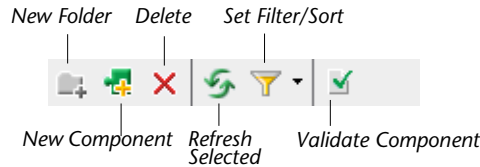


Note: The image above is just an example. When you log into the Quality Center demo for the first time, you may not see the same folders or components in the component tree.

The Business Components module window contains the following key elements used in this tutorial:

- **Components menu bar**—Contains menus with Business Components module commands, from which you can perform module commands, such as those for creating new components and folders in the component tree, copying and pasting, and changing the displayed view.

- **Components toolbar**—Contains buttons for frequently-used commands in the Business Components module.



These commands enable you to create and delete folders and business components, filter and sort the components displayed in the tree, and validate components.

Tip: You can also access many of the menu and toolbar commands by right-clicking in panes or on items in the component tree to access context menus. You can also perform some menu commands using shortcut keys. For more information, refer to the *Mercury Business Process Testing User's Guide*.

- **Component tree**—Enables you to construct and organize the hierarchy of your business component folders and individual components.
- **Business Components Module Tabs**—Selecting a business component in the component tree displays six tabs that provide a complete overview of the component's content, detail its input and output component parameters, describe the component steps, and list the business process test or tests in which it is used.

Exploring the Test Plan Module Window

Business components are combined into business process tests in the Quality Center Test Plan module. In this section you will explore the Business Process Testing elements of the Test Plan module.

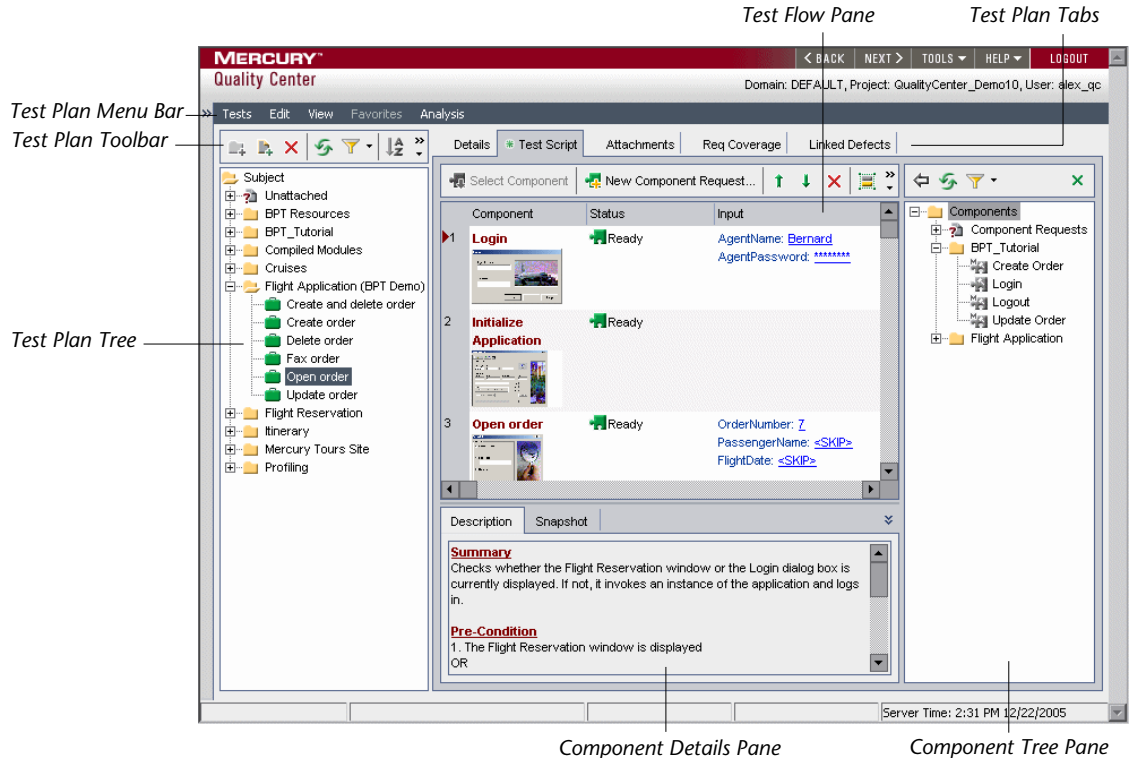
1 Open the Test Plan module.



Click the **Test Plan** module button in the sidebar. The Test Plan module opens.

2 Explore the Business Process Testing elements of the Test Plan module.

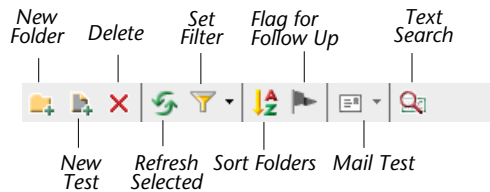
An example of the Test Plan module window is shown below, with the Test Script tab of a business process test selected.



Note: The image above is just an example. When you log into the Quality Center demo for the first time, you may not see the same folders or tests in the test plan tree.

When the Test Plan Tree view is displayed, the Test Plan module window contains the following key elements:

- **Test Plan Menu Bar**—Contains the Tests, Edit, View, and Analysis menus, from which you can perform Test Plan module commands, such as those for creating new business process tests, changing the displayed view, and generating reports.
- **Test Plan Toolbar**—Contains buttons for the commands that are commonly used when creating and modifying the test plan tree, such as creating or deleting business process tests, refreshing the data, and filtering the tree. For more information about these commands, refer to the *Mercury Quality Center User's Guide*.



Tip: You can also access many of the menu and toolbar commands by right-clicking in panes or on items in the test plan tree to access context menus. You can also execute some menu commands using shortcut keys. For more information, refer to the *Mercury Business Process Testing User's Guide*.

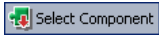
- **Test Plan Tree**—The Test Plan Tree is a graphical representation of your test plan, displaying subject folders and business process tests according to the hierarchical relationship of their purpose within the business process.

Note: You can also view business process tests and other test types in the Test Grid view. In that view, you can see all tests at once, sort and filter them according to your needs, and take advantage of other test management options. For the purposes of this tutorial, you will be working only in the Test Plan Tree view. For more information on the Test Grid view, refer to the *Mercury Quality Center User's Guide*.

- **Test Plan Tabs**—Selecting a business process test in the test plan tree displays five tabs that provide a complete overview of the business process test and its status.

3 Click the Test Script tab.

- **Test Flow Pane**—For each business component in the business process test, the Test Flow pane in the Test Script tab displays the name, status, input and output component parameters, and whether the test will continue if that component fails during the test run.



- **Component Tree Pane**—The Component Tree pane can be optionally displayed on the right side of the Test Script tab by clicking the **Select Component** button in the toolbar. The tree displays the hierarchical structure of all business components in the project.

Now that you are familiar with the main Quality Center windows used for Business Process Testing, take a few minutes to explore their key elements.

In the next section, you will familiarize yourself with the Mercury Flight Reservations sample application.

Exploring the Mercury Flight Reservations Application

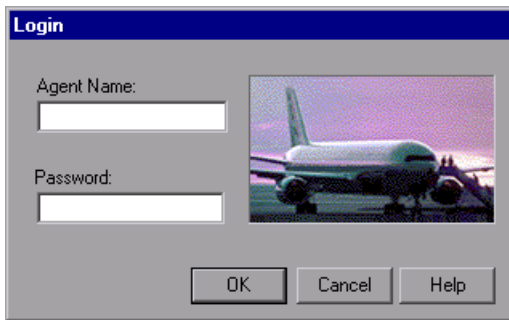
Mercury Flight Reservations is the sample application used throughout this tutorial. It simulates a Windows-based flight reservation service. This section enables you to familiarize yourself with this application before starting the tutorial.

Logging into the Mercury Flight Reservations Application

Before you book, update, or delete a flight in the Mercury Flight Reservations application, you must log in as a user.

To log in to Mercury Flight Reservations:

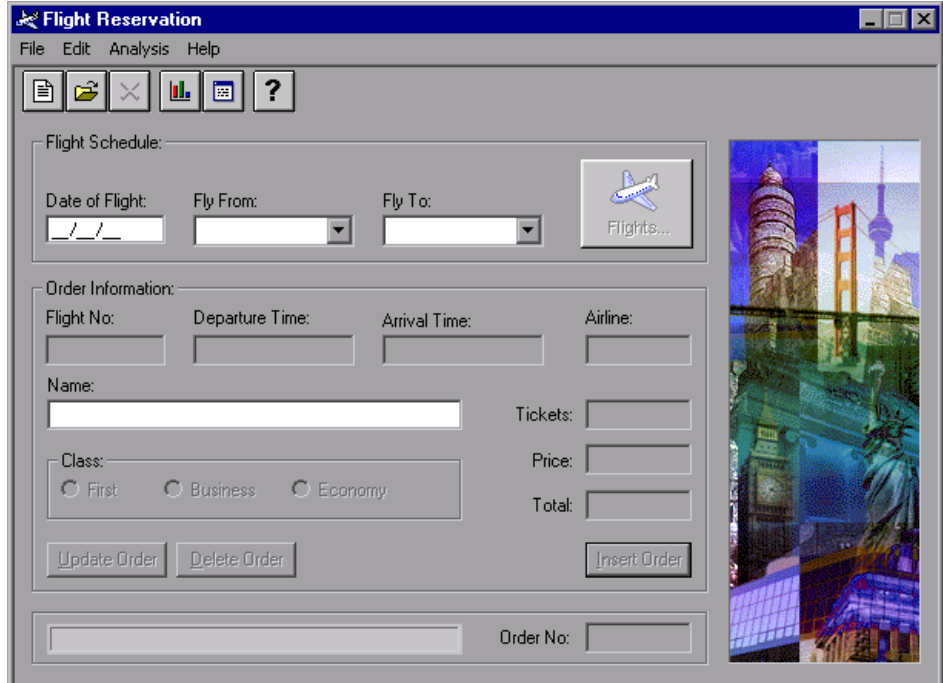
- 1 From the **Start** menu, choose **Programs > QuickTest Professional > Sample Applications > Flight**. The Login dialog box opens.



- 2 Enter a name with at least four characters in the **Agent Name** box. The agent name mercury is used in this tutorial.
- 3 Enter mercury in the **Password** box. The password is not case sensitive.

Note: The characters you enter for the password are displayed as asterisks.

4 Click **OK**. The Flight Reservation window opens.



The screenshot shows the 'Flight Reservation' application window. The title bar is blue with the text 'Flight Reservation' and standard window controls. The menu bar includes 'File', 'Edit', 'Analysis', and 'Help'. Below the menu bar is a toolbar with icons for a document, a folder, a close button, a bar chart, a calendar, and a question mark. The main area is divided into two sections. The 'Flight Schedule:' section contains fields for 'Date of Flight:' (with a date picker), 'Fly From:' (a dropdown menu), and 'Fly To:' (a dropdown menu). To the right of these fields is a button with an airplane icon labeled 'Flights...'. The 'Order Information:' section contains fields for 'Flight No:', 'Departure Time:', 'Arrival Time:', and 'Airline:'. Below these are fields for 'Name:', 'Tickets:', 'Price:', and 'Total:'. There are three radio buttons for 'Class:' labeled 'First', 'Business', and 'Economy'. At the bottom of this section are buttons for 'Update Order', 'Delete Order', and 'Insert Order'. At the very bottom of the window is a field for 'Order No:'. On the right side of the window, there is a vertical strip showing a collage of images including the Golden Gate Bridge, the Statue of Liberty, and other New York City landmarks.

Note: While the Mercury Flight Reservations application loads, the Mercury Flight Reservations splash screen is displayed. Loading the application may take a few seconds.

Explore and experiment with the options and menu commands in this application, including inserting an order, opening an existing order, and updating an existing order. These are the business processes that you will be testing while working in this tutorial.

Note: This tutorial uses the default settings for order number 3 provided in the sample Flight Reservations database. Before you proceed with the other lessons in this tutorial, choose **File > Open Order** in the Flight Reservation window to open the Open Order dialog box, select **Order No.** and enter 3 in the text box. Click **OK** to open order number 3 and ensure **Economy** is selected in the **Class** area of the Flight Reservation dialog box. If **Economy** is not selected, select it and click **Update Order**.

When you have finished exploring the application, choose **File > Exit** to close the application.

Now that you are familiar with the concepts of Business Process Testing, the Business Components and Test Plan modules, and the Mercury Flight Reservations application, you can proceed to Lesson 2, “Creating Business Components.” In Lesson 2, you will learn how to create business components in the Quality Center Business Components module.

2

Creating Business Components

Business components are reusable units that perform a specific task in a business process. Components can be manual or automated. Automated components can be keyword-driven or scripted components.

There are several stages involved in creating a keyword-driven automated component. You begin by creating a manual business component in the Quality Center Business Components module, define the details that make up each component's shell in the Details tab of the component, and add manual steps. At this point, you can add components to business process tests, even if the application is not yet ready to be tested.

Once the application is ready to be tested, you can convert your component to an automated component, attach an application area, and add keyword-driven steps to it.

In this lesson you will learn about:

- Creating Components
- Defining Component Details
- Adding Manual Steps

In later lessons, you will complete the component design process by converting your components to automated components, attaching application areas, and adding keyword-driven steps.

Creating Components

In this section, you create business components for testing the Mercury Flight Reservations application.

1 Open the Business Components module.

If Quality Center is not currently open, open it and log into the Quality Center Demo project, as described in “Starting Business Process Testing” on page 6.



Click the **Business Components** module button in the sidebar. The Business Components module opens.

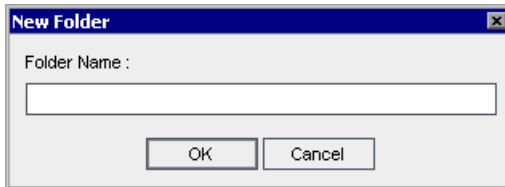
2 Create a folder in which your business components will be stored.

In the component tree, select the **Components** root folder.



Click the **New Folder** button in the tree toolbar.

The New Folder dialog box opens.

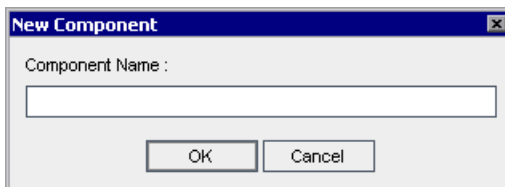


Enter BPT_Tutorial in the **Folder Name** box, and click **OK**. The new folder is added to the component tree.

3 Add the first business component to the folder.



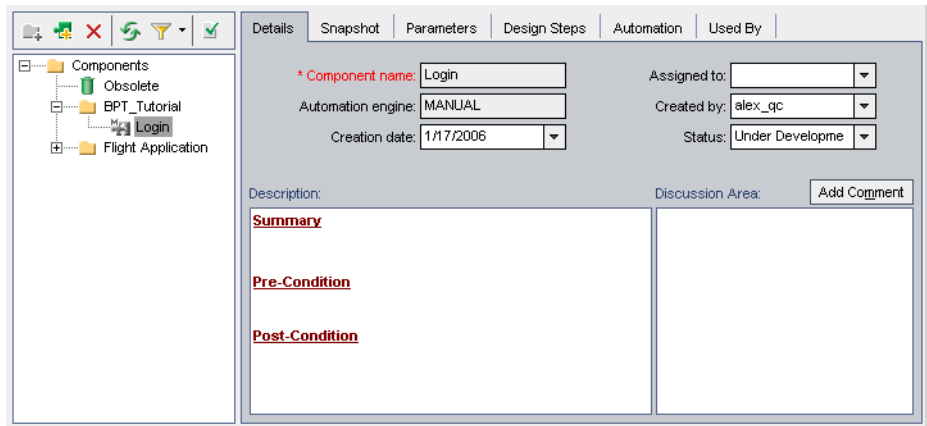
Ensure that the **BPT_Tutorial** folder is selected in the component tree and click the **New Component** button in the tree toolbar. The New Component dialog box opens.



Enter **Login** in the **Component Name** box and click **OK**. The new business component is displayed under the BPT_Tutorial folder in the component tree.

Note that the component icon displays an **M** in the top-left corner, which indicates that it is currently a *manual* component that can only be run manually.

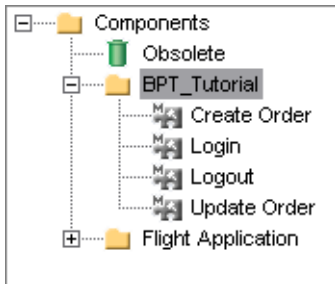
The component tabs for the new **Login** component are displayed on the right of the Quality Center window.



4 Repeat procedure 3 to create the following business components in the Flight_Reservation folder:

- Create Order
- Update Order
- Logout

The components are displayed in the component tree in alphabetical order, as follows:



Defining Component Details

In this section, you define the details that comprise the shell of each business component that you created.

1 Display the Details tab for the component.

Select the **Login** business component in the component tree and click the **Details** tab if it is not currently displayed.

2 Provide the component's general details.

In the **Description** area, use the default section headings to enter a textual description of the **Login** component's purpose, and the pre-condition and post-condition of the application before and after running the component.

- **Summary**—Enter an overall description of the business component's purpose or contents. For example:
Access the Mercury Flight Reservations application, enter name and password, and open the Flight Reservation dialog box.
- **Pre-Condition**—Describe the point in the flow of the application at which the current business component starts. For example:
The Flight Reservations application must be closed.
- **Post-Condition**—Describe the point in the flow of the application at which the current business component ends. For example:
The Flight Reservation dialog box is displayed.

Note: The Details tab provides a comprehensive selection of text editing and formatting commands to assist you in entering the component description. Right-click in the **Description** area to access the commands from the context menu. Choose **Toolbar Visible** from the context menu to display (or hide) commonly used commands in the toolbar.

3 Repeat procedures 1 and 2 to define appropriate details for the other business components.

Set the details of the components according to the table below.

| Component | Summary | Pre-condition | Post-condition |
|---------------------|--|--|--|
| Create Order | Enter flight details and insert an order. | The Flight Reservation window is active. | The Flight Reservation window is active. |
| Update Order | Open an existing order, modify flight details, and update the order. | The Flight Reservation window is active. | The Flight Reservation window is active. |
| Logout | Exit the Mercury Flight Reservations application. | The Flight Reservation window is active. | The application is closed. |

Adding Manual Steps

Now that you have created business components, you can add manual steps for them. Manual steps can be used as instructions for testing the application manually, or they can be used as guidelines for creating automated steps at a later time.

You add manual steps in the Design Steps tab of the Business Components module.

1 Display the Design Steps tab for the component.

Select the **Login** business component in the component tree and click the **Design Steps** tab.

2 Physically perform the first manual step in the Flight Reservations application so that you know how to describe it.

The first step in your login component is to open the Flight Reservations application. Perform this step yourself by finding the executable file in the **<QuickTest Professional installation folder>** and double-clicking it.

3 Enter the first manual step.



Click the **New step** toolbar button. The Component Step editor opens.

Component Step Editor

Step Name:

Description:

Expected Result:

OK Cancel

In the Step Name box, enter a descriptive name, for example, **Launch Flight Application**.

In the **Description** pane, enter a description of the first step of the component. For example:

Open the Mercury Flight Application from C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe

In the **Expected Result** pane, enter the expected result of performing the described step. For example: The Login dialog box opens.

If the step does not produce the expected result when the component is run, the component will fail.



Click the **New Step** button within the Component Step Editor to open a blank second step.

4 Add the remaining manual steps for the Login Component.

Physically perform each of the next steps in the Flight Reservations application (entering a user name, entering a password, clicking **OK** to close the Login dialog box) so that you will know how to describe them.

Then enter a description of the operations you performed in the Component Step Editor by adding a name, description and expected result of each step and clicking the **New Step** button.

5 Close and Save the Login Component.

When you complete the last step in the component, click **OK** in the Component Step Editor. If you are prompted to save your changes, click **Yes**.



Click the **Save** toolbar button to save your steps.



Tip: You must save any changes that you make in the Design Steps tab by clicking the **Save** toolbar button. Since changes you make in this tab are not saved automatically, it is also recommended that you save them periodically.

6 Enter manual steps for the Create Order, Update Order, and Logout components.

Repeat procedures 1 through 5 to enter manual steps for the remaining components.

- The **Create Order** manual steps should include displaying the Flight Reservation window, entering itinerary details, selecting a flight, entering order information, and inserting the order.
- The **Update Order** manual steps should include displaying the Flight Reservation window, opening an order, modifying the seating class to first class, and updating the order.
- The **Logout** manual steps should include displaying the Flight Reservation window, and exiting the application.

When entering manual steps, keep in mind that your first and last steps should reflect the pre-conditions and post-conditions specified when you created the component.

For the purposes of this tutorial, you can be as detailed as desired when entering the manual steps. In a real testing environment, manual steps should include each individual operation that should be performed on the application during the component run as well as additional testing operations such as checking that a button is enabled, making sure an image appears in a particular location, checking how the application responds to invalid data, and so on.

Note: Even if you choose to add less detailed manual steps in the tutorial, it is recommended that you include at least a few manual steps in each component, because some instructions in future sections of the tutorial assume that manual steps (later converted to manual operations) exist.

Now that you are familiar with creating components and defining business component details, you can proceed to Lesson 3, “Creating Business Process Tests.” In Lesson 3, you will learn how to combine the components you have created into a business process test.

3

Creating Business Process Tests

Business process tests are testing scenarios comprised of a serial flow of business components. In this lesson you create a business process test in the Test Plan module of Quality Center, using the components you created in the Business Components module.

Note: For more information about working with the Test Plan module of Quality Center, refer to the *Mercury Quality Center User's Guide*.

In this lesson you will learn about:

- Creating a Business Process Test
- Adding Components to a Business Process Test

Creating a Business Process Test

In this section, you create a new business process test in the test plan tree of the Test Plan module.

1 Open the Test Plan module.



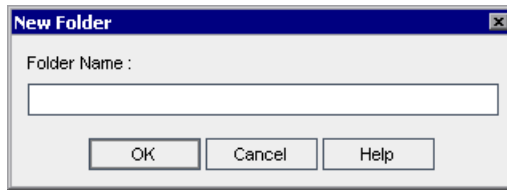
Click the **Test Plan** module button in the Quality Center sidebar. The Test Plan module opens.

2 Create a folder in which your business process tests will be stored.

In the test plan tree, select the **Subject** root folder.



Click the **New Folder** button in the toolbar above the test plan tree. The New Folder dialog box opens.



In the **Folder Name** box, enter BPT_Tutorial, and click **OK**.

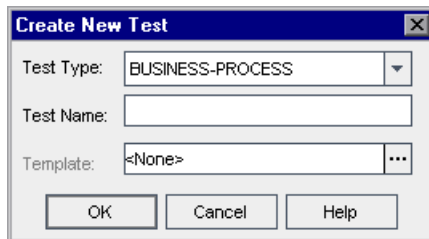
The new folder is displayed as a subfolder of the **Subject** root folder in the test plan tree.

3 Add a new business process test to the folder.



Ensure the new folder is selected in the test plan tree. Click the **New Test** button in the toolbar above the tree. The Create New Test dialog box opens.

From the **Test Type** list, select **BUSINESS-PROCESS**.

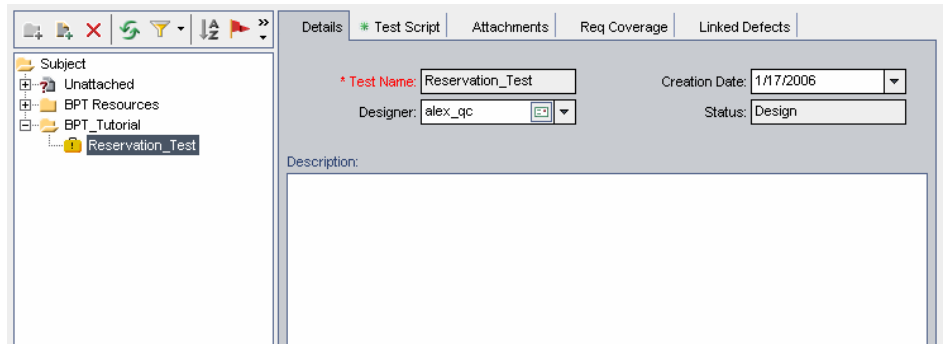


In the **Test Name** box, enter Reservation_Test.

Click **OK**.

Note: If required test fields were defined in the Project Customization window in Quality Center, the Required Fields dialog box opens. Select any value for each of the displayed fields. For more information, refer to the *Mercury Quality Center Administrator's Guide*.

The new business process test is added to the **BPT_Tutorial** folder in the test plan tree. The business process test tabs are displayed on the right of the Quality Center window.



4 Add a test description.

In the Details tab, you can see the **Test Name**, **Creation Date**, **Designer**, and **Status**.

In the **Description** box, enter a description for the business process test. For example: This test verifies the functionality of the Flight Reservations application.

Adding Components to a Business Process Test

In this section, you build the structure and flow of your business process test by selecting the business components you created and adding them to the test. You then configure their run settings, and define what the test will do if one or more of the components fail.

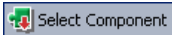
1 Display the available components in the Test Script Tab.

The business components you created in the Business components module are available for testing in the Test Plan module.

Note: All business components in the Business components module, created by you or by others, are available for use. This means that components can be used in multiple business process tests. For example, the **Login** component you created could be used in any test that opens the Mercury Flight Reservations application.

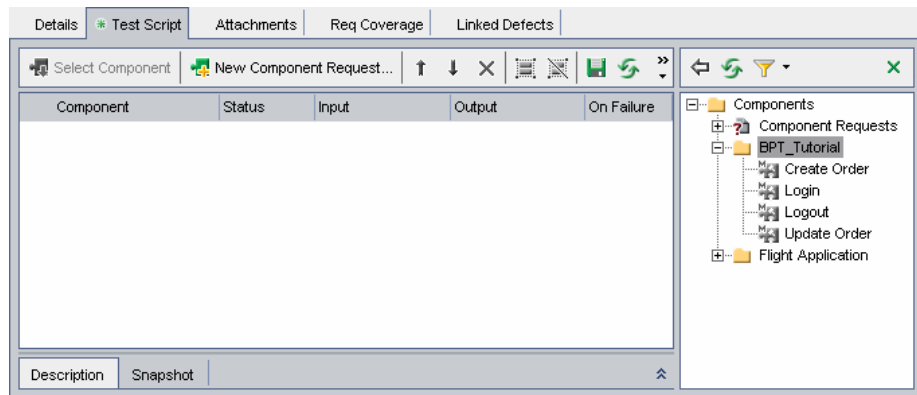
Select **Reservation_Test** in the test plan tree. The test tabs are displayed.

Click the **Test Script** tab. This tab comprises two panes—the Test Flow pane on the left of the tab, and the Component Tree pane on the right. The Component Tree pane contains your business components.



Ensure that the Component Tree pane is displayed. If it is not displayed, click the **Select Component** button in the Test Flow pane toolbar.

In the Component Tree pane, expand the **BPT_Tutorial** folder that contains the business components you created.



Tip: To expand a folder in the Component Tree pane, click the **Expand** symbol to the left of the folder name or double-click the folder.

2 Add components to the test.

Add business components to the business process test according to the following procedures:

Drag the **Login** component from the Component Tree pane and drop it in the Test Flow pane.

Double-click the **Update Order** component in the Component Tree pane to add it to the end of the test.

Drag the **Create Order** component from the Component Tree pane and drop it on top of the **Login** component in the Test Flow pane. This inserts the component directly below the **Login** component.



Ensure that the **Create Order** component is selected in the Test Flow pane. Then select **Logout** in the Component Tree pane and click the **Add Component to Test** button in the toolbar. This adds the component directly below the **Create Order** component.

3 Close the Component Tree pane.



You have added all the required components to your test.

Click the **Close** button in the Component Tree pane toolbar.

4 Arrange the selected components into a logical testing sequence.

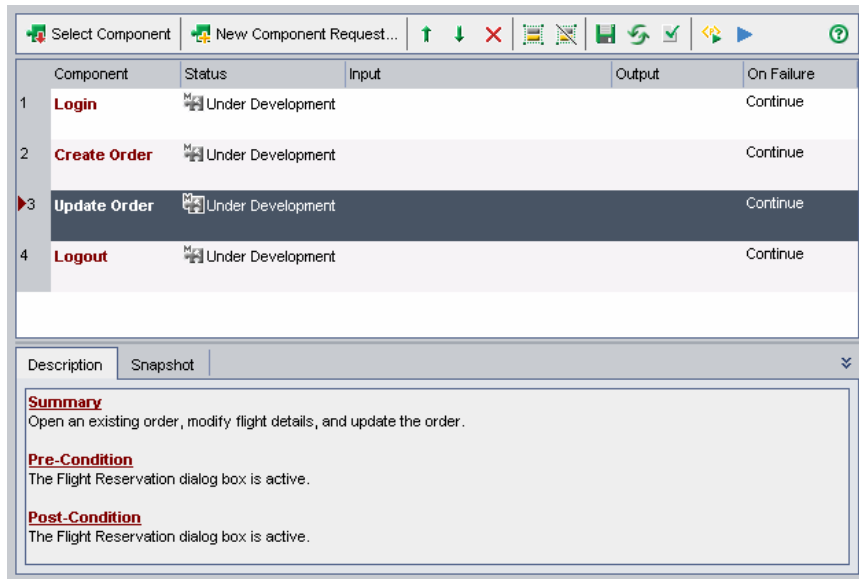


To assist you in arranging your components, you can review the implementation instructions, and the pre- and post-conditions of each component directly from the Test Flow pane. Select a component in the Test Flow pane and click the **Show** button at the bottom right of the pane. The Description tab displays the information you entered in the Details tab of the component in the Business Components module.

Update Order is currently positioned as the last component in the test. When the test is run, the **Logout** component must be the last component.



Select **Update Order** in the Test Script pane and click the **Up** button in the toolbar once, to move **Update Order** above the **Logout** component.



5 Save the Business Process Test.



Click the **Save** toolbar button to save your new business process test.

Defining Failure Conditions




If the failure of a particular component means that later components in the test cannot possibly run successfully, or that there is simply no point to continue the test run after such a failure, you can instruct the business process test to end at that point.

You will now define the condition that your test run will end if the **Login** component fails to open the Flight Reservations application.



- 1 In the Test Flow pane of the Test Script tab, click in the **On Failure** column for the **Login** component. A down arrow is displayed. (You may have to scroll to the right to view this column.)

- 2 Click the down arrow and select **Exit** from the list. This defines that your business process test will end if the **Login** business component fails.

| | Component | Status | Input | Output | On Failure |
|---|--------------|---|-------|--------|------------|
| 1 | Login |  Under Development | | | Exit |
| 2 | Create Order |  Under Development | | | Exit |
| 3 | Update Order |  Under Development | | | Continue |
| 4 | Logout |  Under Development | | | Continue |

Note: By default, **Continue** is defined for each business component when it is added to a business process test. This means that the test will run the next component even if the specific component fails.

You have now finished the first phase of business process testing. Having created business components with manual steps and combining those components into business process tests, you could already perform manual tests on your application.

You are also ready to begin the next phase of business process testing: automating your components. The next two lessons will teach you how to convert your manual components to automated components and add keyword-driven steps to them.






This tutorial does not describe manual business process test runs. For more information, refer to the *Mercury Business Process Testing User's Guide*.

4

Automating Components

Before you can add automated steps in a keyword-driven business component, you must convert the component to an automated component. You then attach an application area that includes all the objects of the component contained in the application as well as any special functions or recovery scenarios that you may need to access.

When you create the steps of a business component, the objects in the application area are displayed for selection in a repository tree, which provides a hierarchical view of all the available objects in the application. For example, an object repository tree in a Flight Reservation business component could contain:

- Dialog boxes 
- Text boxes 
- Buttons 
- List Boxes 
- Radio buttons 

It is important to note that when the business component runs, the context of each step in the component must be set by an earlier step or steps. This means that for an operation on an object to be performed successfully, the object must be currently displayed.

In this lesson you will learn about:

- Converting Components to Automated Components
- Selecting Application Areas

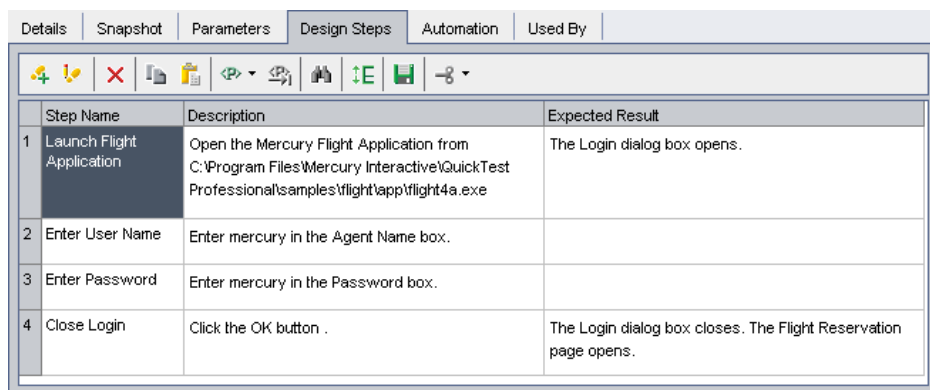
Converting Components to Automated Components

In this section, you will access the Design Steps tab for each manual business component you created in Lesson 2 and convert them to automated components.

Note: You cannot modify the manual steps in your components after they are automated.

1 Display the Design Steps tab for the Login component.

In the Business Components module, select the **Login** component in the component tree, and click the **Design Steps** tab. The Design Steps tab opens.



| Step Name | Description | Expected Result |
|-----------------------------|---|---|
| 1 Launch Flight Application | Open the Mercury Flight Application from C:\Program Files\Mercury Interactive\QuickTest Professionals\samples\flight\app\flight4a.exe | The Login dialog box opens. |
| 2 Enter User Name | Enter mercury in the Agent Name box. | |
| 3 Enter Password | Enter mercury in the Password box. | |
| 4 Close Login | Click the OK button . | The Login dialog box closes. The Flight Reservation page opens. |

Note: Predefined project resources are created the first time you click the **Design Steps** tab for a component in a new project. These resources are displayed in the BPT Resources folder in the Test Plan module. During this process, the Creating Predefined Business Process Testing Resources progress bar is displayed. This may take a few seconds to complete.

2 Automate the component.

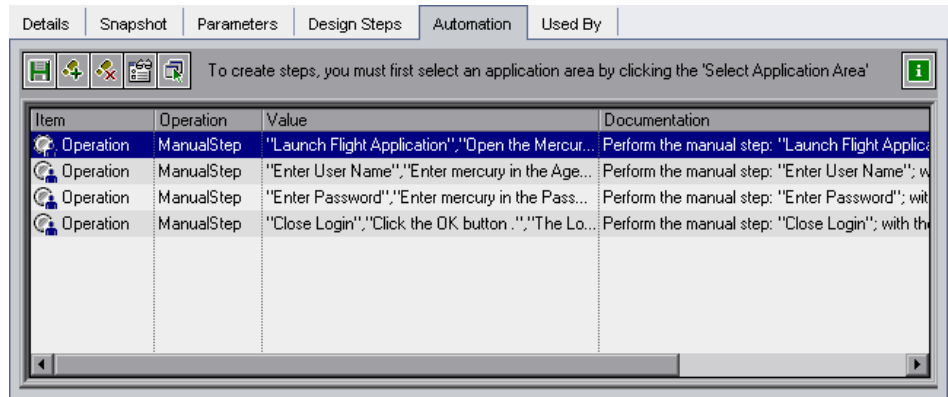


Click the **Automate component** toolbar button.

Select **QuickTest Keyword-Driven** from the list and click **Yes** in the warning message. The component is converted to an automated QuickTest component.

Click **OK** in the information message and then click the **Automation** tab.

The steps appear as manual step operations in the Automation tab.



Tip: You can select which columns to display in the Automation tab, specify their order, and set the number of columns that remain stationary when you scroll horizontally. For more information about setting keyword view options, refer to the *Mercury Business Process Testing User's Guide*.

3 Automate the remaining components.

Repeat steps 1 and 2 to automate the **Create Order**, **Update Order**, and **Logout** components.

Selecting Application Areas

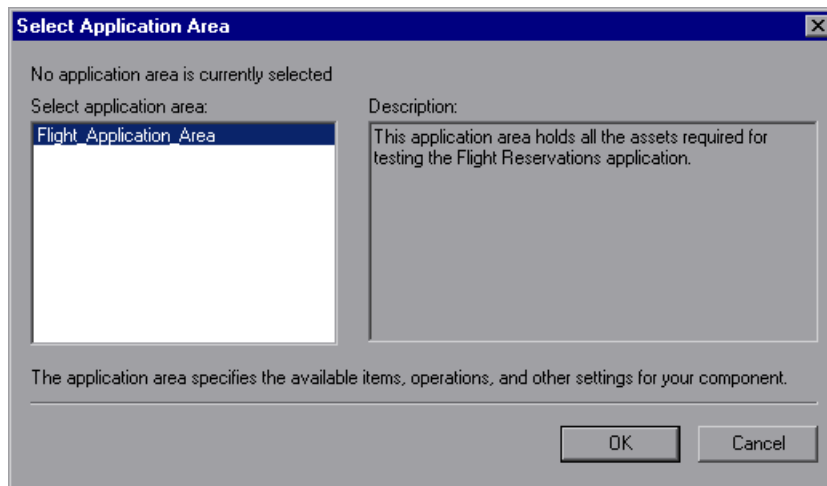
When you work with keyword-driven components, you must assign an application area to each component. The application area determines the objects and operations that are available for your component as well as other default settings such as associated add-ins, recovery scenarios, the Windows-based applications on which your component can run, and more.

The application area is usually created by the automation expert in consultation with the Subject Matter Experts who will be using it. In this tutorial, a pre-made application area has already been prepared for you.

1 Select the application area for the Login component.

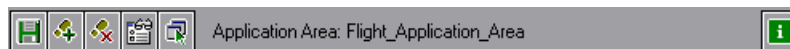


Click the **Select Application Area** button in the Automation tab toolbar. The Select Application Area dialog box opens.



Ensure that **Flight_Application_Area** is the currently selected application area for the business component.

Click **OK**. The Select Application Area dialog box closes. The name of the associated application area (**Flight_Application_Area**) is displayed in the toolbar of the Automation tab.



2 Save the component.



Click the **Save** button in the toolbar to save the **Login** business component.

Tip: Changes that you make in this tab are saved automatically only when you select another tab in the component, another location in the component tree, or another module. It is recommended that you save your changes by clicking the **Save** button periodically while you work.

3 Select the **Flight_Application_Area** application area for the other components.

Different components can use different application areas even if they will be part of the same business process test. However, in this tutorial, the **Create Order**, **Update Order** and **Logout** business components all use **Flight_Application_Area**.

Open the **Automation** tab of each of the other business components in turn and select **Flight_Application_Area** for each one.

Remember to save each component after you make any change.

Now that you are familiar with automating components and attaching application areas to business components, you can proceed to Lesson 5, “Adding Keyword-Driven Component Steps.” In Lesson 5, you will learn how to create the sequential keyword-driven steps that comprise your business components.

5

Adding Keyword-Driven Component Steps

In Lesson 2, you created manual steps that described the operations that each component should perform. In Lesson 4, you converted your manual steps to manual step operations. In this lesson you will use these operations as a guide for adding automated keyword-driven steps to your business components.

You add automated steps by selecting specific objects from the object repository, created as part of the application area of each component.

Each step you create in the business component is displayed as a row in the Automation tab of the Business Components module. You then select the object (if applicable) and operation to be performed and define any required operation values.

The steps are automatically documented in the Automation tab by Quality Center, providing a plain-language textual description of what the step does.

In order for sequential business components to run successfully in a business process test, remember that the state of the application at the end of one component (its post-conditions) must match the required state of the application at the beginning of the next component (the subsequent component's pre-conditions).

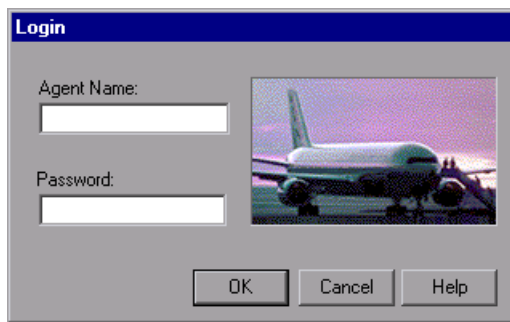
In addition, if you want a business component to run multiple iterations during the test, the application's end state must match its state at the beginning of the component. This means that the first step of the component can be repeated immediately after the last step of the previous component iteration.

In this lesson you will learn about:

- Creating Automated Steps for the Login Component
- Creating Automated Steps for the Create Order Component
- Creating Automated Steps for the Update Order Component
- Creating Automated Steps for the Logout Component

Creating Automated Steps for the Login Component

In this section you identify and create automated keyword-driven steps in the **Login** business component. This component logs the user into the Flight Reservations application.



As you can see from the information you entered previously in the Details and Design Steps tabs, logging in to the application requires the following individual actions:

- Opening the Flight Reservations application
- Entering the agent's name and password in the Login dialog box
- Closing the dialog box

The pre-condition for this business component is that the application is not currently running on your computer.

1 Open the Login component in the Business Components module.

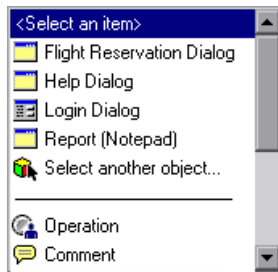
Select the **Login** business component in the component tree and click the **Automation** tab if it is not currently displayed.

2 Open the Flight Reservations application and open the Login dialog box.

This step provides the path to the Flight Reservations application and activates the Login dialog box.



Highlight the manual operation for opening the flight application and click the **Add Step** button in the Automation tab toolbar. A new row is added below the manual operation and the **Select an item** menu opens.



Opening an application is a standalone operation as opposed to one performed on a particular object. Select **Operation** from the list. The **Operation** item is added to the row.



Click in the **Operation** column of the new **Operation** step. A down arrow is displayed.

Click the arrow, and select **OpenApp** from the list.

Click in the **Value** column, and then enter the location of the Flight Reservations application:

<QuickTest Professional installation folder>\samples\flight\app\flight4a.exe

(By default, this path is: C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe)

3 Display the Login dialog box objects.



In the Automation tab, select the step containing the manual operation for entering a user name and click the **Add Step** button in the toolbar. A new row is added below the manual operation and the **Select an item** menu opens.



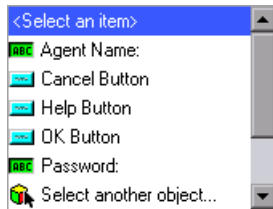
Select **Login Dialog**. The Login Dialog step is added as a row in the tab, and the default operation for the item, **Activate**, is automatically inserted in the **Operation** column. This step ensures that the parent Login dialog box object is active, and that the child objects within the dialog box are displayed for selection in the **Select an item** menu. Subsequent steps can then be created to perform operations on the child objects.

4 Add the other login steps to the component.

You can now add the **Agent Name** and **Password** text box objects as steps in the business component.



Select the **Login Dialog** step in the Automation tab and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Note: The **Select an item** menu displays objects in the object repository that are siblings (at the same level), or children (at levels below), of the object in the selected row.



Select **Agent Name**. The **Agent Name** text box object is added as a row in the tab. The default operation for the item, in this case, **Set**, is automatically inserted in the **Operation** column.

In the **Value** column, enter mercury.



Select the step containing the manual operation for entering a password and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Password**. The **Password** text box object is added as a row in the tab. The default operation for the item, **Set**, is automatically inserted in the **Operation** column.

In the Value column, enter mercury.

Note: The **Operation** list also offers a **SetSecure** operation. This operation enables you to enter password values in an encrypted format. You can generate the encrypted value using the Password Encoder tool supplies with QuickTest. However, for the purposes of this tutorial, you are leaving the password unencrypted by using the **Set** operation.



Select the step containing the manual operation for clicking the **OK** button, and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **OK Button**. The **OK Button** object is added as a row in the Automation tab.

The default operation for the item, **Click**, is automatically inserted in the **Operation** column.

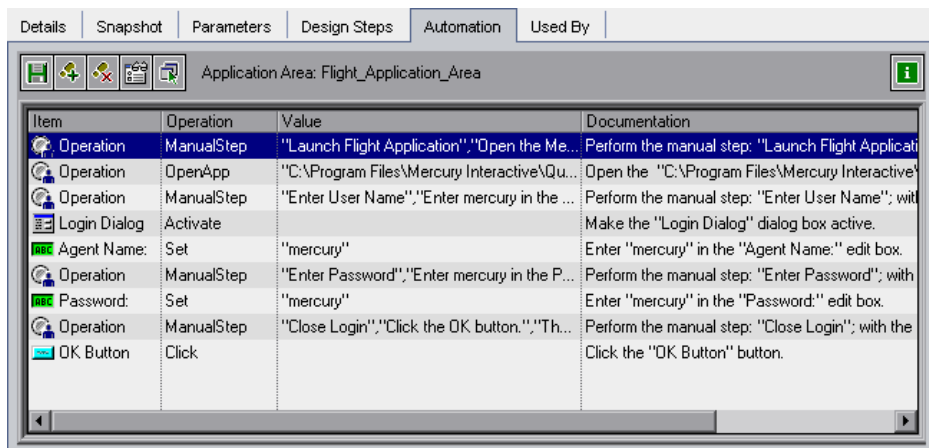
This final step opens the Flight Reservation dialog box. This is the pre-condition for running the next component, **Create Order**.

5 Review your component.

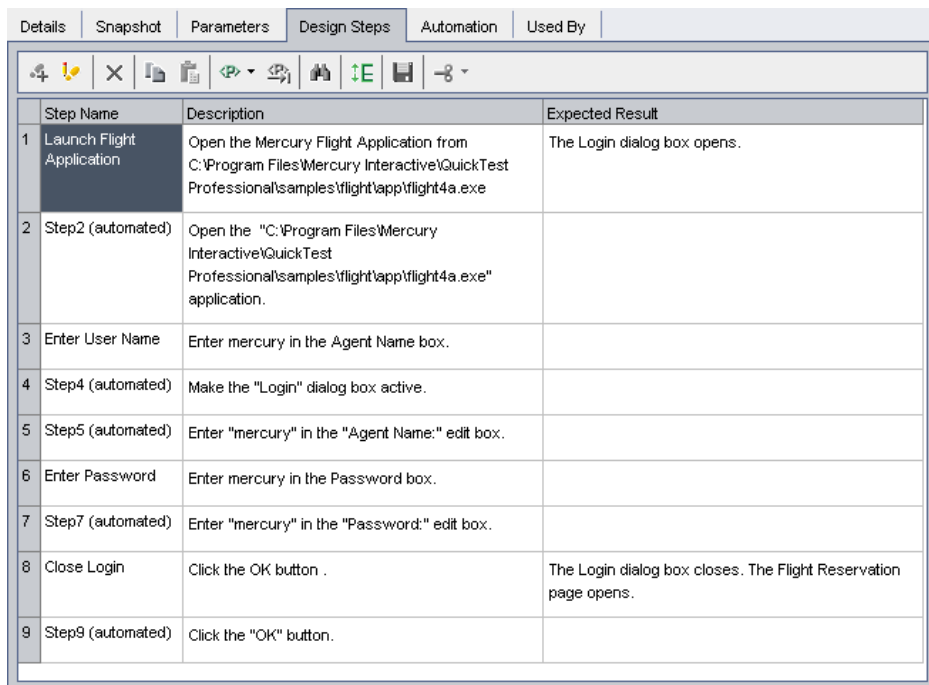
Look at the documentation column in the Automation tab. (You may need to scroll to the right to see the entire column). Review the documentation that was automatically generated for each step you entered to ensure that the steps make sense and follow a logical order.

Lesson 5 • Adding Keyword-Driven Component Steps

The Automation tab for the Login business component should appear similar to the following.



Click the **Design Steps** tab. Each of your original manual steps is followed by the equivalent automated step.



6 Save the component.



Click the **Save** button in the toolbar to save the **Login** component.

Deleting Manual Step Operations

The manual steps that you originally created in the Design Steps tab provided summary information or implementation instructions about the component and have no actual effect on the automated test run. Since your manual steps were mainly guides to what needed to be implemented in the automated steps, they are no longer necessary.

Therefore, after you have implemented the automated steps in your component, you may want to delete some or all of the manual step descriptions (now manual step operations) from the component to make it easier to read.

1 Delete the manual steps.

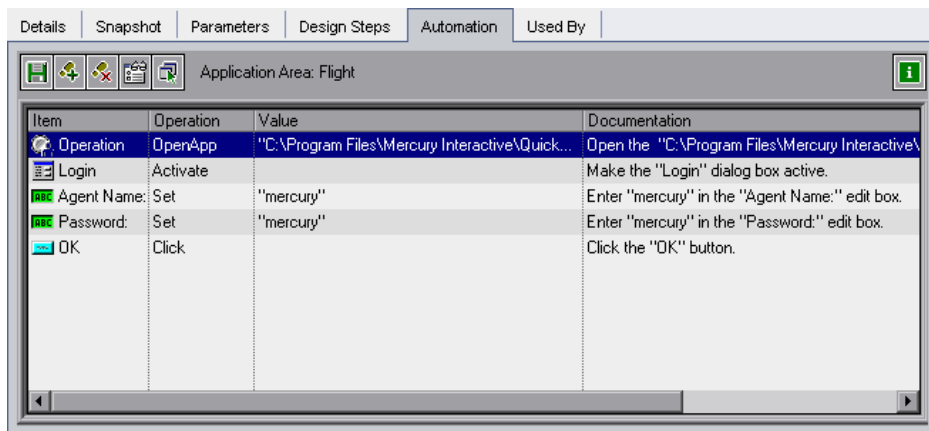


In the Automation tab, highlight the first manual step operation in the **Login** component and click the **Delete Step** button. Click **Yes** on the warning message. Repeat for each manual operation in your component that you want to delete.

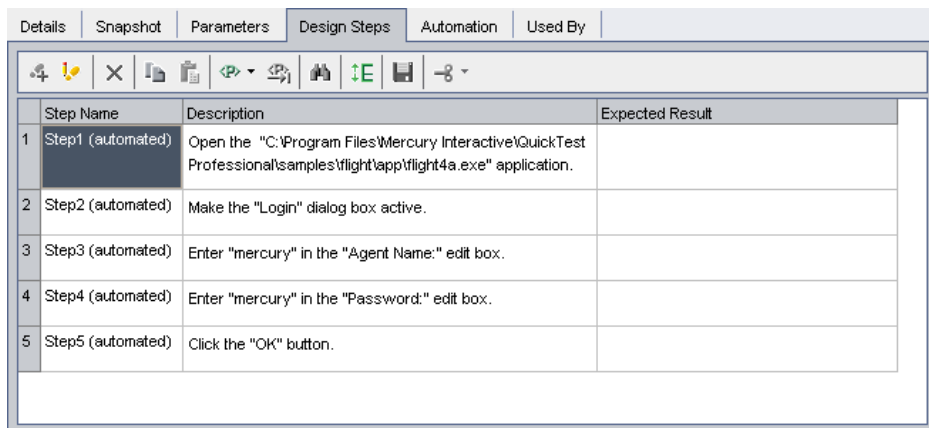
2 Save the component.



Click the **Save** button in the Automation tab toolbar to save the **Login** component.

3 Review the automated keyword-driven steps in the Automation tab.**4 Review the automated steps in the Design Steps tab.**

The steps in this tab are now read-only and can only be changed by modifying (or adding to) the steps in the Automation tab.



Creating Automated Steps for the Create Order Component

In this section you identify and create automated keyword-driven steps in the **Create Order** business component. This component reserves a specific flight for a passenger and creates an order number in the system.

The steps in this component use the Flight Reservation dialog box:

Creating the order requires the following individual actions:

- Displaying the Flight Reservation objects
- Entering flight schedule requirements
- Selecting an available flight
- Entering order information
- Creating the order in the system
- Retrieving the generated order number and store it for later use

The pre-condition for this business component is that the Flight Reservation window is currently displayed. This condition is answered by the **Login** component that runs before the **Create Order** component in the test and opens the application.

1 Open the Create Order component in the Business Components module.

Select the **Create Order** business component in the component tree and click the **Automation** tab if it is not currently displayed.

2 Display the Flight Reservation objects and clear any dialog box values.

The first step ensures that the Flight Reservation dialog box is active. The next step clears any data that may have been entered by a previous iteration of this component or another component that may perform operations on this dialog box earlier in a test.



Select the first manual operation in the component and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Flight Reservation Dialog** from the list. The **Flight Reservation Dialog** object is added as a row in the tab, and the default operation for the item, **Activate**, is automatically inserted in the **Operation** column.



Select the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Scroll down to **Toolbar.New Order Button** in the list (the list is in alphabetical order) and select it. Accept the default **Click** operation. This step clicks the **New Order** button and clears any data in the Flight Reservations window, making it ready to accept new order details. This is particularly important as it enables this business component to be iterated.

3 Enter flight schedule requirements.

You will now create the component steps that enter the information required to select a specific flight to be entered—the required flight date, the starting point, and the destination.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **OrderInformation.Date of Flight**. This item enables the required departure date to be inserted in the **Date of Flight** text box. Accept the default **Set** operation. In the **Value** column, enter 01/20/07 (January 20, 2007).

Repeat the steps in this procedure to add the following steps:

| Item | Operation | Value |
|--------------------------|-----------|---------------|
| Flight Schedule.Fly From | Select | San Francisco |
| Flight Schedule.Fly To | Select | Zurich |

4 Display the Flights table.

Now you will insert the component step that opens the **Flights** table containing the available flights according to the flight schedule requirements.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Flight Schedule.Flights**. Accept the default **Click** operation.

5 Select an available flight.

The next steps open the Flights dialog box and select an available flight.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Flights Table Dialog** from the list. Accept the default **Activate** operation.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **OK Button**. Accept the default **Click** operation. This step clicks the **OK** button to accept the first available flight listed in the **Flights** table.

6 Display the Flight Reservation dialog box.

The objects required for the following steps are located in the Flight Reservation dialog box. This step makes the Flight Reservation dialog box active again.

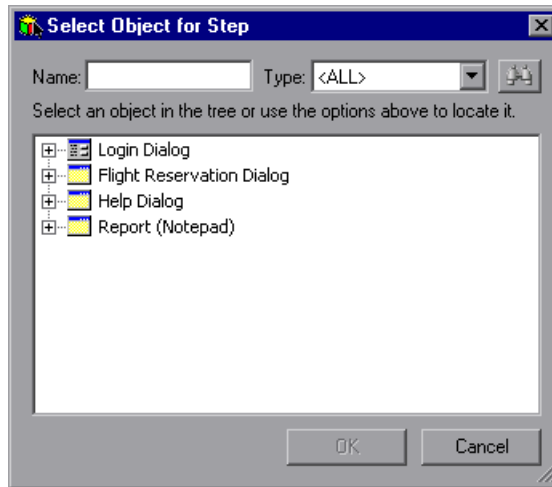


Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.

Flight Reservation Dialog is not a sibling or child object of the **OK Button** object, and is therefore not displayed in the **Select an item** list.



Select **Select another object** from the list. The Select Object for Step dialog box opens.



Select **Flight Reservation Dialog** in the Object repository tree and click **OK**. The Select Object for Step dialog box closes and the Flight Reservation dialog box step is added with the default **Activate** operation.

7 Enter order information.

This step enables the passenger name and class details to be defined for the selected flight.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Order Information.Name**. This item enables the passenger name to be inserted into the **Name** text box. Accept the default **Set** operation and enter J. Jones in the **Value** column.

8 Create the order in the system.

This step reserves the flight according to the specified details and creates an order number in the system.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Order Information.Insert Order Button**. Accept the default **Click** operation.

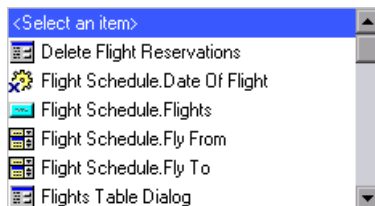
9 Retrieve the generated order number and store it for later use.

When a new order is inserted in the Flight Reservation dialog box, it generates an order number. This order number can potentially be useful for many other testing tasks in other components. Therefore, it makes sense to retrieve this value and store it.

In this step, you will store the retrieved value in a local variable. In Lesson 7, “Working with Parameters,” you will learn how to store this value in a component parameter so that other components can use the retrieved value.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.





Select the **Order No** text box. The default operation for the item, **Set**, is automatically inserted in the **Operation** column. Change the operation to **GetValue** by selecting it from the **Operation** list.



Click in the **Output** column, and then click the **Specify parameter for output** button.

In the Output Options dialog box, accept the **Local Parameter** output type. Enter the parameter name, **Temp_OrderNo_Out**, and for the description, enter **Temporary output parameter for storing the retrieved order number**.

Click **OK** to close the Output Options dialog box.

10 Delete any unnecessary manual steps.

If desired, you can delete the manual steps as described in “Deleting Manual Step Operations” on page 45.

11 Review the auto-documentation for the component.

Read through the statements in the Documentation column to ensure that the steps make sense and follow a logical order.

12 Save the component.



Click the **Save** button in the toolbar to save the **Create Order** component.

The steps in the Automation tab for the **Create Order** business component should appear similar to the following:

| Item | Operation | Value | Output | Documentation |
|---------------------------------------|-----------|-----------------|------------------------------------|---|
| Flight Reservation Dialog | Activate | | | Make the "Flight Reservation Dialog" window active. |
| Toolbar.New Order Button | Click | | | Click the "Toolbar.New Order Button" button. |
| Order Information.Date of Flight | Set | "01/20/07" | | Type "012007" in the "Flight Schedule.Date of Flight" text box. |
| Flight Schedule.Fly From | Select | "San Francisco" | | Select the "San Francisco" item from the "Flight Schedule.Fly From" list. |
| Flight Schedule.Fly To | Select | "Zurich" | | Select the "Zurich" item from the "Flight Schedule.Fly To" list. |
| Flight Schedule.Flights | Click | | | Click the "Flight Schedule.Flights" button. |
| Flights Table Dialog | Activate | | | Make the "Flights Table Dialog" dialog box active. |
| OK Button | Click | | | Click the "OK Button" button. |
| Flight Reservation Dialog | Activate | | | Make the "Flight Reservation Dialog" window active. |
| Order Information.Name | Set | "J. Jones" | | Enter "J. Jones" in the "Order Information.Name" text box. |
| Order Information.Insert Order Button | Click | | | Click the "Order Information.Insert Order Button" button. |
| Order No | GetValue | | LocalParameter("Temp_OrderNo_Out") | Return the "Order No" edit box value. Store it in the "Temp_OrderNo_Out" local parameter. |

Creating Automated Steps for the Update Order Component

The order you created in the **Create Order** component reserved an Economy Class seat for passenger J. Jones. In this section you identify and create the automated keyword-driven steps in the Update Order business component. This component reopens this order and upgrades the reservation to First Class.

Updating the order requires the following individual actions:

- Locating the order to be updated
- Opening the order
- Making the required changes
- Updating the order

The pre-condition for this business component is that the Flight Reservation window is currently displayed.

1 Open the Update Order component in the Business Components module.

Select the **Update Order** business component in the component tree and click the **Automation** tab if it is not currently displayed.

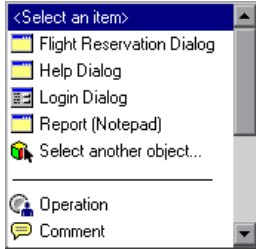
2 Access the Open Order dialog box.

These steps capture the order number of the order you created in the **Create Order** component.



Select the first manual operation in the component and click the **Add Step** button in the Automation tab toolbar.

A new row is added to the tab and the **Select an item** menu opens.



Select **Flight Reservation Dialog** from the list and accept the default **Activate** operation. Adding this step ensures that the Flight Reservation dialog box is active.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.

Select **Menu**. Accept the default **Select** operation. In the **Value** column enter the following, exactly as shown: `File;Open Order...`

This data is case sensitive.

3 Open the order to be updated.

To make your Update Order component more reusable, you can set the order number that you open as a parameter. This enables each test that uses the component to assign a different value to use.

In these steps you use a local parameter (variable) value. In Lesson 7, “Working with Parameters,” you will change it to a component parameter so that you can retrieve the order number value from another component.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Open Order Dialog**. Accept the default **Activate** operation. Adding this step ensures that the Open Order dialog box is active.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select the **Order No** check box. Accept the default **Set** operation. In the **Value** column, enter ON. This step places a check mark in the **Order No** check box in the Open Order dialog box, which enables the number of the required order to be inserted.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Order Number Edit**. Accept the default **Set** operation.



In the **Value** column, click the **Configure the value** button. In the Value Configuration Options dialog box, select the **Parameter** radio button. **Local Parameter** is displayed in the corresponding text box. In the **Name** box, enter Temp_OrderNo_In. In the **Value** box, enter 3. In the **Description** box, enter Temporary input parameter representing the retrieved order number.

In Lesson 7, “Working with Parameters,” you will change this local parameter to a component input parameter that is defined to retrieve the value from output value step of the **Create Order** component.

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

4 Close the Open Order dialog box.

This step closes the Open Order dialog box and displays the order details in the Flight Reservation dialog box.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **OK Button**. Accept the default **Click** operation.

In the following steps, you make a change to the original reservation you created.

5 Display the Flight Reservation dialog box.

The objects required for the following steps are located in the Flight Reservation dialog box. This step ensures that the Flight Reservation dialog box is active.

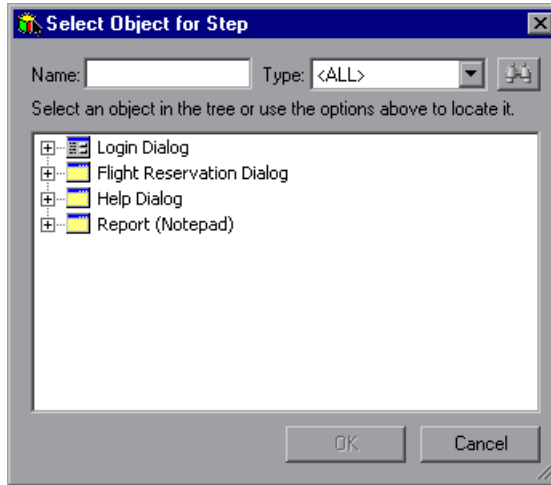


Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.

Flight Reservation Dialog is not a sibling or child object of the **Open Order Dialog** object, and is therefore not displayed in the **Select an item** list.



Select **Select another object** from the list. The Select Object for Step dialog box opens.



Select **Flight Reservation Dialog** in the Object repository tree and click **OK**. The Select Object for Step dialog box closes.

6 Update the order.

These steps update the order by changing the passenger class.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Order Information.Class.First**. Accept the default **Set** operation. This step selects the **First Class** radio button.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Order Information.Update Order Button**. Accept the default **Click** operation. This step clicks the **Update Order** button, which updates the order with the changed data.

7 Delete any unnecessary manual operations.

If desired, you can delete the manual operations, as described in “Deleting Manual Step Operations” on page 45.

8 Save the component.



Click the **Save** button on the Automation tab toolbar to save the Update Order component.

The steps in the Automation tab for the Update Order business component should appear similar to the following:

| Item | Operation | Value | Documentation |
|-------------------------------|-----------|-----------------------------------|---|
| Flight Reservation Dialog | Activate | | Make the "Flight Reservation Dialog" window active. |
| Menu | Select | "File;Open Order..." | Select item "File;Open Order..." from the "Menu". |
| Open Order Dialog | Activate | | Make the "Open Order Dialog" dialog box active. |
| Order No. | Set | "ON" | Set the state of the "Order No." check box to "On". |
| Order Number Edit | Set | LocalParameter("Temp_OrderNo_In") | Enter the value of the "Temp_OrderNo_In" local parameter. |
| OK Button | Click | | Click the "OK Button" button. |
| Flight Reservation Dialog | Activate | | Make the "Flight Reservation Dialog" window active. |
| Order Information.Class.First | Set | | Select the "Order Information.Class.First" radio button. |
| Order Information.Update ... | Click | | Click the "Order Information.Update Order Button". |

Creating Automated Steps for the Logout Component

In this section you identify and create the automated keyword-driven steps in the **Logout** business component. This component exits the Flight Reservations application.

You exit the application using the File menu in the Flight Reservation dialog box. Exiting the Flight Reservations application requires the following individual operations:

- Displaying the Flight Reservation dialog box
- Selecting the Exit option

The pre-condition for this business component is that the Flight Reservation window is currently displayed.

1 Open the Logout component in the Business Components module.

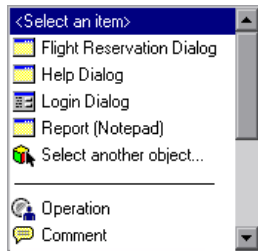
Select the **Logout** business component in the component tree and click the **Automation** tab if it is not currently displayed.

2 Display the Flight Reservation dialog box.

Adding this step ensures that the objects in the Flight Reservation dialog box are displayed for selection in the **Select an item** menu.



Select the first manual operation and click the **Add Step** button in the Automation tab toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Flight Reservation Dialog** from the list. Accept the default **Activate** operation.

3 Select Exit from the File menu.



Select the next manual operation or the step you just inserted and click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Menu**. Accept the default **Select** operation. In the **Value** column, enter the following, exactly as shown: `File;Exit`

This data is case sensitive.

4 Delete any unnecessary manual steps.

If desired, you can delete the manual operations as described in “Deleting Manual Step Operations” on page 45.

5 Save the component.



Click the **Save** button on the Automation tab toolbar to save the **Logout** component.

The steps in the Automation tab for the completed **Logout** business component should appear similar to the following:

| Item | Operation | Value | Documentation |
|---------------------------|-----------|-------------|---|
| Flight Reservation Dialog | Activate | | Make the "Flight Reservation Dialog" window active. |
| Menu | Select | "File;Exit" | Select item "File;Exit" from the "Menu" menu. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Your business components are now ready to run. In the next lesson, you will perform a debug run on your business process test to ensure that all your components run successfully.

6

Debugging Business Process Tests

You check for any problems arising from the content of automated business components, and their combination in a business process test, by running the test in debug mode.

Debugging ensures that your business process test runs properly, and locates any possible errors that may have occurred during the building of the test. For example, you may want to check that the business components are in a logical order in the test, and that the pre-conditions and post-conditions required by each component are correctly applied.

When the debug run begins, Quality Center opens QuickTest Professional directly from the Test Plan module.

(If the first component in the test has any add-ins associated with it, those add-ins are loaded when QuickTest opens. However, for the components in this tutorial, no add-ins are necessary.)

QuickTest then performs each of the automated steps that you created in your business components.

When the test is complete, you can view a summarized report that shows which of the business components passed or failed the test.

In this lesson, you check that the individual business components you have created run together successfully in the business process test.

Notes:

You must have QuickTest Professional version 9.0 installed on your computer to run a business process test in the Test Plan module.

In addition, before you run a business process test on a host computer, you must ensure that QuickTest Professional on the host allows Quality Center to run tests. Open QuickTest on the host computer, choose **Tools > Options**, click the **Run** tab, and select **Allow other Mercury products to run tests and components**.

Debugging Your Business Process Test

Now that you have created a business process test (Lesson 3, “Creating Business Process Tests”), automated your components (Lesson 4, “Automating Components”), and added steps to your components (Lesson 5, “Adding Keyword-Driven Component Steps”), you are ready to debug your test.

1 Open the Test Plan module.



Click the **Test Plan** module button in the Quality Center sidebar. The Test Plan module opens.

2 Display the components in the test.

Ensure that the **Reservation_Test** business process test is selected in the test plan tree, and the **Test Script** tab is displayed.

Note: Before you run a business process test, you must ensure that all the necessary applications for the test are set up to meet the pre-conditions required by the first business component. In this case, for example, the **Login** component opens the Flight Reservations application, so you must close any instance of the application that may be open. You should also ensure that QuickTest is not open on your computer with any unsaved test or component.

3 Set all components to exit on failure.

While editing and debugging your test, it is a good idea to set all components to exit the test run if one of the components fail. That way, if a component fails, you do not have to wait for all other components to run before trying to solve the problem. Once your test is finalized and ready to be run in a test set, you can return the exit conditions to their previous settings.

In the **On Failure** column, select **Exit** for each of the components.

4 Save the component.



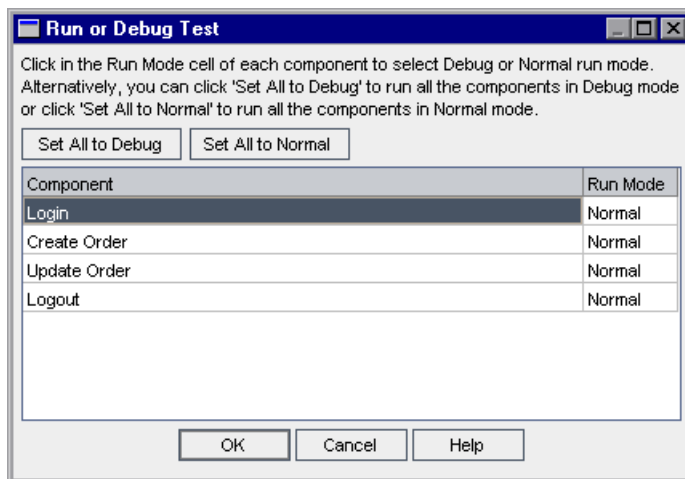
Click the **Save** button in the toolbar to save your changes.

5 Run the test.



Click the **Run or Debug Test** button in the Test Script toolbar.

The Run or Debug Test dialog box opens, which lists all the business components that make up the selected business process test.



When you set a component to **Normal** mode, the component runs from start to finish without pausing and continues immediately to the next component in the test. When you run a component in **Debug** mode, a breakpoint is added that causes the business process test to pause before running the first step in the selected business component. For more information about selecting **Debug** mode or **Normal** mode in the Run or Debug Test dialog box, refer to the *Mercury Business Process Testing User's Guide*.

For the purposes of this lesson, you are checking that the test as a whole runs smoothly. Therefore, leave all components set to **Normal** mode.

Click **OK** to close the Run or Debug Test dialog box. QuickTest Professional opens on your computer and runs the first business component in the business process test.

Note: If QuickTest displays a permissions error when you begin the run, ensure that QuickTest Professional on the host allows Quality Center to run tests, as described in the note on page 62.

6 View the progress of the test.

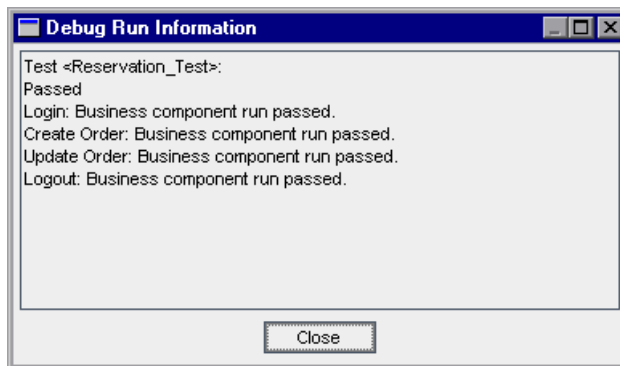
Toggle between the Test Plan module, QuickTest, and the Flight Reservations application by clicking the relevant buttons on the task bar.

- In the Test Plan module, the status of the business process test and the name of the business component currently being run are shown below the toolbar in the Test Script tab, for example, **Running (Login)**.
- In QuickTest Professional, the status of the business process test being run, for example **Running**, or **Ready**, is shown in the status bar at the bottom-right of the QuickTest window.

When one business component finishes running, it closes and the next component in the business process test opens in QuickTest.

7 Complete the run.

When the debug run is complete, Quality Center imports the results from QuickTest and displays summary information in the Debug Run Information dialog box in the Test Plan module.



You can see that all your components ran successfully. Click **Close** to complete the test run.

Note: The debug run updated the passenger class of order number 3 in the Mercury Flight Reservations sample database. To run this test again, or to prepare the data for the next user, open the Mercury Flight Reservations application and choose **File > Open Order** to open the Open Order dialog box. Select **Order No** and enter 3 in the text box. Click **OK** to open order number 3, select **Economy** in the Class area of the Flight Reservation dialog box and click **Update Order**.

7

Working with Parameters

In this lesson, you will learn how to define and use input and output *component parameters*. Input component parameters enable you to retrieve data from a source outside the steps of the component. The values may be supplied by the test containing the component, the test set that runs a particular test, or data retrieved from another component in the test. Output component parameters are data values that are retrieved from a component step in one business component (the *source* component) and can be passed as input parameters to a subsequent component (the *target* component) in the business process test flow.

Parameterization greatly increases the power and flexibility of your component or business process test by letting you check how your application performs the same operations with multiple sets of data.

This lesson describes:

- Defining an Output Component Parameter for the Create Order Component
- Defining Input Parameters for the Create Order and Update Order Components
- Creating the Delete Order Component and Adding It to Your Test
- Iterating Components
- Debugging the Parameterized Test
- Grouping Components

Defining an Output Component Parameter for the Create Order Component

In this section, you define the name, value type, and description of a new output parameter for the **Create Order** business component. You then modify the **GetValue** step in the component to return and store the current order number in the new parameter. The value of this parameter will be the number of the order that is updated during the test run.

1 Open the Create Order component in the Business Components module.



Click the **Business Components** module button in the sidebar. The Business Components module opens.

Select the **Create Order** business component in the component tree and click the **Parameters** tab.

2 Create a new output parameter for the Create Order component.



In the **Output** pane toolbar, click the **New Output Parameter** button. The New Parameter dialog box opens:

Enter **Order_No_Out** in the **Parameter Name** box and click **OK**.

A line for the new output component parameter is added in the Output parameter list.

| Output | | |
|---|------------|-------------|
| <div> New Delete </div> | | |
| Parameter Name | Value Type | Description |
| Order_No_Out | String | |
| | | |

String is entered automatically in the **Value Type** cell.

3 Define the parameter details.

Select the **Value Type** cell for the component parameter and select **Number** from the list.

In the **Description** cell, enter The retrieved reservation order number as a description of the component parameter.

4 Store the reservation order number in the Order_No component parameter.

Click the **Automation** tab.

Click in the **Output** column of the **GetValue** step. Two buttons are displayed to the right of the parameter name.



Click the **Specify parameter for output** button to open the Output Options dialog box.

Select **Component parameter** in the **Output Types** box. Because only one component parameter has been defined, **Order_No_Out** is automatically displayed in the **Name** box.



Click the down arrow to display the description you entered in step 3, in read-only format.

Click **OK**. The Output Options dialog box closes.

5 Save the changes to your component.



Click the **Save** button on the Automation tab toolbar.

Defining Input Parameters for the Create Order and Update Order Components

In this section you will insert two input component parameters. The first will accept data provided at the test level so that each test using the component can provide different data. The second parameter will retrieve its data from the output value of a previous component.

Using Test Data as Input

In this section you will create a new input parameter in which you can define various destination cities in the **Create Order** component. This component parameter replaces the fixed value, Zurich, that you defined previously.

1 Display the component Parameter tab.



Click the **Business Components** module button in the sidebar. The Business Components module opens.

Select the **Create Order** business component in the component tree, and click the **Parameters** tab.

2 Create a new input parameter for the Create Order component.



In the **Input** pane toolbar, click the **New Input Parameter button**. The New Parameter dialog box opens:

A screenshot of the "New Parameter" dialog box. It has a title bar with "New Parameter" and a close button. Inside, there is a label "Parameter Name :" followed by a text input field. At the bottom, there are two buttons: "OK" and "Cancel".

Enter **Destination** in the **Parameter Name** box and click **OK**. A line for the new input component parameter is added in the **Input** parameter list. **String** is entered automatically in the **Value Type** cell.

| Parameter Name | Value Type | Default Value | Description |
|----------------|------------|---------------|-------------|
| Destination | String | | |
| | | | |

In the **Default Value** cell, enter **Zurich**. This is the value that will be used if no value is provided by the test using the component.

In the **Description** cell, enter a description of the component parameter's purpose, or other descriptive text, such as the valid values for the parameter. For example, **Destination cities. Possible values: Denver, Frankfurt, London, Los Angeles, Paris, Portland, San Francisco, Seattle, Sydney, Zurich.**

3 Change the destination city from a fixed value to a component parameter.

Click the **Automation** tab.




In the **Flight Schedule.Fly To** row, click in the **Value** column. The **Configure the value** button is displayed. Click the button. The Value Configuration Options dialog box opens.



Click the **Parameter** radio button and ensure that **Component parameter** is selected. Click the down arrow next to the parameter name to display the description of the parameter you created in step 2.

Click **OK**. The Value Configuration Options dialog box closes.

Note: If you click in the **Value** cell of a component parameter in the Automation tab, the component parameter icon  is displayed.

4 Save the changes to your component.



Click the **Save** button on the Automation tab toolbar.

Click the **Test Plan** module button in the sidebar to open the Test Plan module. Select **Reservation_Test** in the test plan tree, and click the **Test Script** tab if it is not currently displayed. The input parameter you created, and a link, are displayed in the **Input** column.

Input Parameter
Link

| | Component | Status | Input | Output | On Failure |
|---|--------------|-------------------|-------------------------------------|--------------|------------|
| 1 | Login | Under Development | | | Exit |
| 2 | Create Order | Under Development | Destination: Zurich | Order_No_Out | Exit |
| 3 | Update Order | Under Development | | | Exit |
| 4 | Logout | Under Development | | | Exit |



Tip: If you do not see the new parameter, click the **Refresh** button in the Test Flow pane toolbar.

When you move the cursor over the link, the cursor changes to a pointing hand. Later in this lesson, you will click this link to configure other parameter values.

Using Output Parameters as Input

Earlier in this lesson, you created a step that stores the generated order number from the **Create Order** component in an output component parameter. Now you will modify the step in the **Update Order** component that opens an order so that it can retrieve the order number from the **Create Order** output parameter.

1 Open the Update Order component in the Business Components module.



Click the **Business Components** module button in the sidebar. The Business Components module opens.

Select the **Update Order** business component in the component tree and click the **Parameters** tab.

2 Create a new input parameter for the Update Order component.

In the **Input** pane toolbar, click the **New Input Parameter** button. The New Parameter dialog box opens. Enter `Order_Number_In` in the **Parameter Name** box and click **OK**. A line for the new input component parameter is added in the **Input** parameter list.

String is entered automatically in the **Value Type** cell. Change the value type to **Number** by selecting it from the list.

In the **Default Value** cell, enter 3. This is the value that will be used if no value is provided by the test using the component. In the **Description** cell, enter The flight reservation order to open.

3 Parameterize the Order Number Edit Step.

Click the **Automation** tab and locate the **Order Number Edit** step.

Click in the **Value** column, and then click the **Configure the value** button. The Value Configuration Options dialog box opens and shows the local parameter that you defined in Lesson 5, “Adding Keyword-Driven Component Steps.”



From the **Parameter** list, select **Component parameter**. Click the down arrow next to the parameter name to display the description of the parameter you created in step 2.

Click **OK**. The Value Configuration Options dialog box closes and the value cell displays a component parameter icon.

Later, you will configure this parameter to use the output value generated by the **Create Order** component.

4 Save the changes to your component.

Click the **Save** button on the Automation tab toolbar.

Creating the Delete Order Component and Adding It to Your Test

In this section, you create a new business component for your business process test, whose properties and steps are based on an existing component. This component deletes the order created and updated during the test run.

Deleting an order requires the following individual actions:

- Locating the order to be deleted
- Opening the order
- Deleting the order

These steps are similar to the steps that you used in the Update Order business component. Quality Center enables you to quickly create a new component by copying an existing component. You can then use the settings and steps of the original component as the basis for the new component.

You create the new Delete Order component based on the existing Update Order component, and then change the steps of the new component as required.

1 Make a copy of the Update Order component.

Select **Update Order** in the component tree, right-click, and choose **Copy**.

Right-click the **BPT_Tutorial** folder in the component tree and choose **Paste**. A duplicated component name warning is displayed. Click **OK**. A copy of the Update Order component, named **Update Order_Copy_1**, is pasted in the component tree.

Right-click the new component and choose **Rename**. Enter **Delete Order** as the name of the component and click on another location to save the new name.

Click the **Automation** tab for the **Delete Order** component if it is not already displayed. Note that this component uses the same input parameter as the Update Order component to open the order number that was originally generated in the **Create Order** component. Only the steps specific to updating the order itself need to be changed.

2 Delete the step not required by Delete Order.

The **Order Information.Class.First** radio button step is not required in the Delete Order component.



Select the **Order Information.Class.First** step and click the **Delete Step** button in the toolbar. A warning message is displayed. Click **Yes**.

3 Change the Update Order Number step to a Delete Order step.



In this component, the Delete Order button must be clicked instead of the Update Order button. Select **Order Information.Update Order Button** in the **Item** column. A down arrow is displayed. Click the arrow and select **Order Information.Delete Order Button** from the list to change the button item that will be clicked in the step. Keep the default **Click** operation.

4 Add additional steps.

Deleting an order in the Flight Reservations application opens a confirmation message dialog box. You must add the following additional steps to the Delete Order component to respond to the message and close the dialog box.

Select the empty row after the last step. The **Select an item** menu opens.



Select **Delete Flight Reservations** from the list. This step ensures that the Flight Reservations warning dialog box is active. The **Delete Flight Reservations** object is added as a row in the Automation tab, and the default operation for the item, **Activate**, is automatically inserted in the **Operation** column.



Click the **Add Step** button in the toolbar. A new row is added to the tab and the **Select an item** menu opens.



Select **Yes Button**. Accept the default **Click** operation. This step clicks the **Yes** button to confirm the deleted reservation.

5 Save the Delete Order business component.



Click the **Save** button on the Automation tab toolbar to save the Delete Order component.

The Automation tab for the completed Delete Order business component should appear similar to the following:

| Item | Operation | Value | Documentation |
|---------------------------------------|-----------|-----------------------------------|---|
| Flight Reservation Dialog | Activate | | Make the "Flight Reservation Dialog" window ac |
| Menu | Select | "File;Open Order..." | Select item "File;Open Order..." from the "Menu" |
| Open Order Dialog | Activate | | Make the "Open Order Dialog" dialog box active |
| Order No. | Set | "ON" | Set the state of the "Order No." check box to "O |
| Order Number Edit | Set | LocalParameter("Temp_OrderNo_In") | Enter <the value of the 'Temp_OrderNo_In' local |
| OK Button | Click | | Click the "OK Button" button. |
| Flight Reservation Dialog | Activate | | Make the "Flight Reservation Dialog" window ac |
| Order Information.Delete Order Button | Click | | Click the "Order Information.Delete Order Button" |
| Delete Flight Reservations | Activate | | Make the "Delete Flight Reservations" dialog bo |
| Yes Button | Click | | Click the "Yes Button" button. |

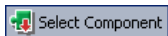
6 Open the test in Test Plan.



Click the **Test Plan** module button in the sidebar to open the Test Plan module.

If the **Reservation_Test** business process test is not currently displayed, select **Reservation_Test** in the test plan tree, and click the **Test Script** tab.

7 Display the Component Tree pane in the Test Script tab.



Click the **Select Component** button in the Test Flow pane toolbar.

In the Component Tree pane, expand the **BPT_Tutorial** folder that contains your business components.

8 Add the Delete Order component to the test.

Select the **Update Order** component in the Test Flow pane. Double-click or drag the **Delete Order** business component in the Component Tree pane to add it to the test below the **Update Order** component and above the **Logout** component.

In the **On Failure** column, change the failure condition to **Exit**.

9 Save the component.

Click the **Save** button in the toolbar to save your changes.

10 Close the Component Tree pane.

Click the **Close** button in the Component Tree pane toolbar.

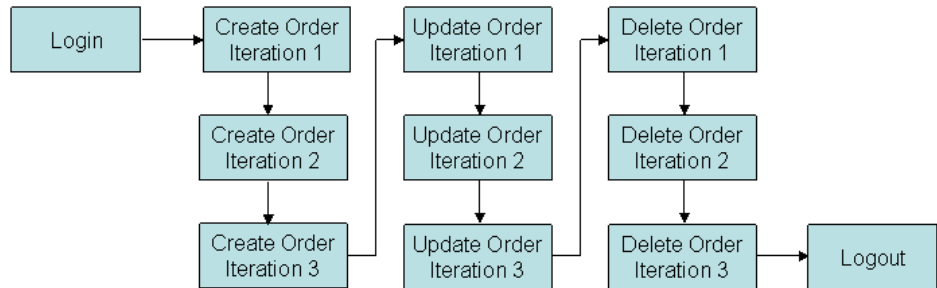
Now that you have parameterized steps in your test, you are ready to configure the values of these parameters and define the iterations for your test.

Iterating Components

Iterations are the number of times that an individual business component, or group of components, runs within a single test run, or the number of times that an entire business process test runs within a test set.

In this section you will configure the **Create Order**, **Update Order**, and **Delete Order** business components to each run three iterations.

The flow of business components and iterations is illustrated as follows:



After opening the Flight Reservations application and logging in, the business process test runs the **Create Order** component three times consecutively, each time creating an order with a different destination. The **Update Order** component then runs three times, and updates each of the created orders. Then the **Delete Order** iteration deletes each of the orders. The test then exits the Mercury Flight Reservations application.

Iterating Using Test Data

Earlier in this lesson, you parameterized the **Fly To** step of the **Create Order** component. Now you will configure your test so that the step will use data defined by the test for each iteration.

1 Display the Test in the Test Plan module.

Ensure that the **Reservation_Test** business process test is selected in the test plan tree of the **Test Plan** module, and click the **Test Script** tab.

2 Configure various destination cities.

Click the **Destination** input parameter link in the **Input** column of the **Create Order** component.

The Component Iterations dialog box opens.

Component <Create Order> Iterations

Specify the parameter values that the business component will use for each iteration during a business process test run. Alternatively, select the Output parameter check box to use an output parameter value from a previous business component as input for this component.

Add Iteration
 Delete Iteration
 Select Iterations
 Import...
 Export...

| Parameter Value | Destination (string) |
|-----------------|---|
| Zurich | <input type="checkbox"/> Output of previous component |
| | |

Parameter <Destination> description:

Destination cities. Possible values: Denver, Frankfurt, London, Los Angeles, Paris, Portland, San Francisco, Seattle, Sydney, Zurich.

OK Cancel Help

Select the **Destination** parameter column. Zurich is displayed in the Parameter Value cell.

Note: The value type defined for the component parameter, for example, **String** or **Number**, is displayed at the top of the column. You can only configure a value that matches the value type for the component parameter.



Click the **Add Iteration** button in the toolbar twice. Two additional iteration rows are added to the Component Iterations dialog box. The parameter value Zurich from the previous row is automatically duplicated in the new rows.

In the **Iteration #1** and **Iteration #2** cells, overwrite the value Zurich with the values Paris and Seattle respectively.

Tip: You can press the TAB or Up and Down arrow keys on your keyboard to move from one iteration row to another.

Click **OK**. The number of defined iterations for the **Create Order** component is displayed in the **Input** column of the Test Script tab.

| Select Component New Component Request... | | | | |
|--|--------------|-------------------|---|------------|
| | Component | Status | Input | On Failure |
| 1 | Login | Under Development | | Exit |
| 2 | Create Order | Under Development | 3 Component Iterations Destination | Exit |
| 3 | Update Order | Under Development | Order_No_In: 3 | Exit |
| 4 | Delete Order | Under Development | Order_No_In: 3 | Exit |
| 5 | Logout | Under Development | | Exit |

Iterating Using Output Values from Previous Components

Earlier in this lesson, you parameterized the **Order Number Edit** step of the **Update Order** component. Now you will configure your test so that the step will use data generated from a previous component.

1 Display the Test in the Test Plan module.

Ensure that the **Reservation_Test** business process test is selected in the test plan tree of the **Test Plan** module, and click the **Test Script** tab.

2 Configure the Order Number input value.

Click the **Order_Number_In** input parameter link in the **Input** column of the **Update Order** component.

The Component Iterations dialog box opens.

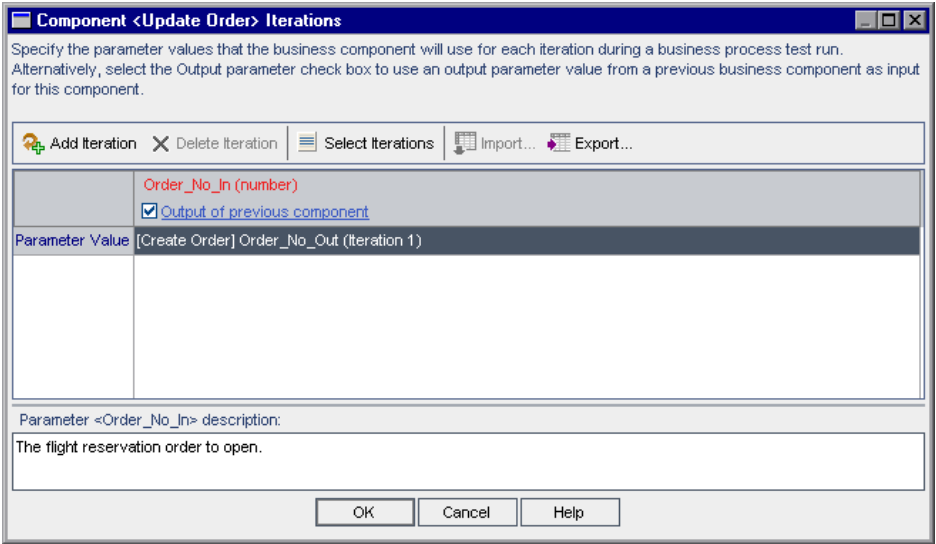
3 Use data from a previous component as the input for this parameter.

Select the **Output of prev component** check box. The Select Output Parameter dialog box opens. Only one output parameter has been defined in previous components in the test so it is automatically selected.

Click **OK**. An iterations mismatch confirmation message is displayed. This indicates that the number of iterations currently defined for your **Update Order** component (1 iteration) does not match the number of iterations currently defined for the **Create Order** component from which you are retrieving the data (3 iterations).

Click **Yes** to close the confirmation message.

The Component Iterations dialog box indicates that the parameter will use the value generated from the first iteration of the **Order_No_Out** parameter in the **Create Order** component. The input parameter is shown in red to indicate an iterations error.



Click **OK** in the dialog box to close it. The iterations error is shown in red in the Test Script tab.

| | | | | | |
|---|--------------|---------------------------------------|--|--------------|------|
| 2 | Create Order | Under Development | 3 Component iterations Destination | Order_No_Out | Exit |
| 3 | Update Order | Under Development Iterations error | Order_No_In: [Create Order] Order_No_Out | | Exit |

4 Add iterations to your Update Order component.

To make the number of iterations in the **Update Order** component match those of the **Create Order** component, you must add two more iterations.

Click the Iterations link of the **Update Order** component to open the Component Iterations dialog box.



Click the **Add Iteration** button in the toolbar twice. Two additional iteration rows are added to the Component Iterations dialog box.





The parameter values show that each iteration of the Update Order component will use the value of the **Order_No_Out** parameter from the corresponding **Create Order** component.

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

Note that the iterations error is no longer shown in the Test Script tab.

5 Add iterations to your Delete Order component.

Your **Delete Order** component uses the same input parameter as the **Update Order** component. Repeat steps 2 through 4 to add three iterations to the **Delete Order** component, each using the **Order_No_Out** parameter from the corresponding **Create Order** component.

| | Component | Status | Input | Output | On Failure |
|-----|--------------|---|---|--------------|------------|
| 1 | Login |  Under Development | | | Exit |
| 2 | Create Order |  Under Development | 3 Component iterations Destination | Order_No_Out | Exit |
| 3 | Update Order |  Under Development | 3 Component iterations Order_No_In | | Exit |
| ▶ 4 | Delete Order |  Under Development | 3 Component iterations Order_No_In | | Exit |
| 5 | Logout |  Under Development | | | Exit |

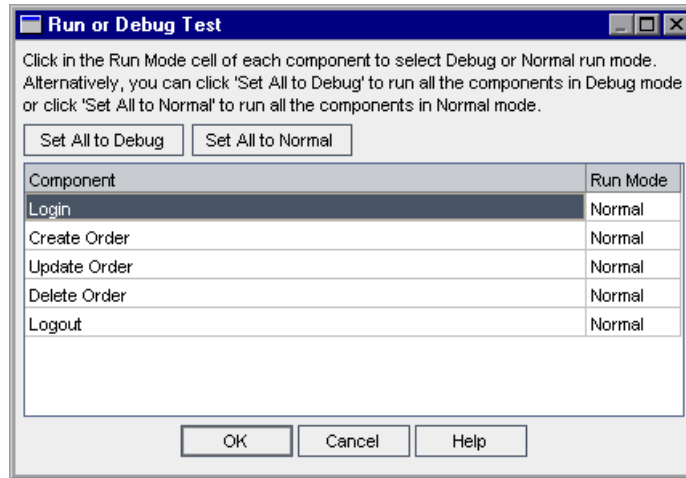
Debugging the Parameterized Test

You have made several modifications to your test. Before moving forward, it is a good idea to run your test again to confirm that it runs successfully.

1 Run the test.

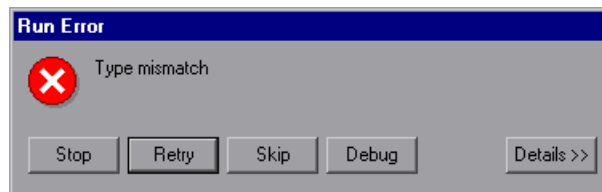


In the Test Flow pane of the Test Script tab, click the **Run or Debug Test** button in the toolbar. The Run or Debug Test dialog box opens.



You are confirming that the test as a whole still runs smoothly after your changes. Therefore, leave all components set to **Normal** mode.

Click **OK** to close the Run or Debug Test dialog box. QuickTest Professional opens on your computer and begins to run the test. However, at the end of the **Create Order** component, a QuickTest error message is displayed.

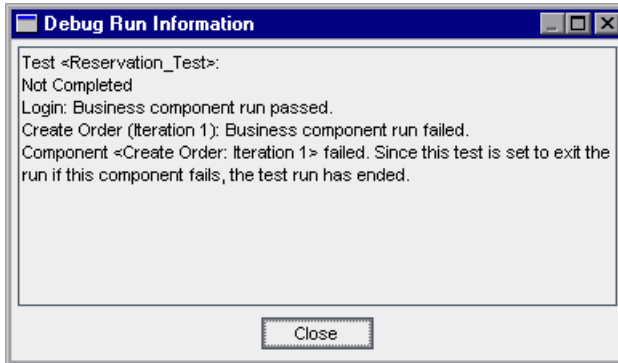


Click **Details** to view the cause of the error.

2 Stop the component and test run.

Click **Stop** in the Run Error dialog box. QuickTest stops running the component and, because the test is set to exit if the component fails, the test run also ends.

When the run stops, the component remains open in QuickTest. Display Quality Center by clicking the Microsoft Internet Explorer icon in the task bar. The Debug Run Information dialog box indicates that the run failed.



Click **Close**.

Exit the Flight Reservations application.

3 Analyze the problem.

The QuickTest error message indicated that it was unable to retrieve a number for the `Order_No_Out` parameter. The reason for this is that QuickTest tries to perform the **GetValue** step immediately after it clicks the **Insert Order** button in the previous step.

Therefore, it attempts to retrieve the value before the Mercury Flight Reservations application has finished inserting the order and at that time, the **Order No** text box is empty.

When the Mercury Flight Reservations application is inserting the order, the dialog box is not active. To ensure that the **Insert Order** operation is complete before the **GetValue** step is performed, you can add a step that activates the Flight Reservation dialog box. This will cause QuickTest to wait until it can identify the dialog box (it can only identify active dialog boxes) and will give the flight application time to complete the order transaction.

4 Modify the component to solve the problem.



Click the **Business Components** module button in the sidebar and select the **Create Order** component. Click the **Automation** tab if it is not already displayed.



Select the **Order Information.Insert Order Button** step and click the **Add Step** button in the toolbar. A new row is added and the **Select an item** menu opens.

The **Flight Reservation Dialog** is not a sibling or child of the **Order Information.Insert Order Button** item, and is therefore not displayed in the **Select an item** list.



Select **Select another object** from the list. The Select Object for Step dialog box opens. Select **Flight Reservation Dialog** in the Object repository tree and click **OK**. The Select Object for Step dialog box closes.

Accept the default **Activate** operation for the step.

5 Save the component.



Click the **Save** button in the Automation tab toolbar to save the Update Order component.

6 Set the test to run in debug mode.

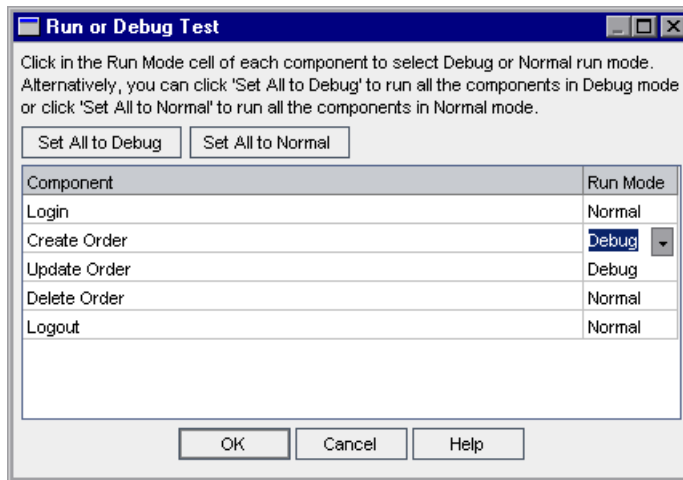


Click the **Test Plan** module button in the sidebar. Ensure that the **Reservation_Test** business process test is selected in the test plan tree, and click the **Test Script** tab.



Click the **Run or Debug Test** button in the toolbar. The Run or Debug Test dialog box opens.

The test run failed between the **Create Order** and **Update Order** components. In the **Run Mode** column of the **Create Order** and **Update Order** components, select **Debug**.



7 Begin the debug run.



Click **OK** in the Run or Debug Test dialog box to begin the debug run. QuickTest performs the **Login** component steps and opens the **Create Order** component. Then it pauses at the first step in the component. A breakpoint icon appears to the left of the first step.

Ensure that QuickTest and the Mercury Flight Reservations application are positioned so that you can see both applications simultaneously.



In the QuickTest Test toolbar, click the **Run** button. QuickTest runs the steps in the component.

Note that the arrow in the QuickTest margin now pauses for several seconds next to the **Flight Reservation Dialog.Activate** step while it waits for the dialog box to become available so that it can perform the **Activate** operation, and that it proceeds to the **Order No.GetValue** step only after the Insert Order operation is complete.

Because you are running several iterations of the components, the **Create Order** component opens twice more, each time stopping at the breakpoint at the beginning of the component. Click **Run** twice more to complete the **Create Order** component iterations.

During each iteration of the **Create Order** component, note the order numbers that the Flight Reservation dialog box displays in the **Order No** box. These are the numbers that your component is retrieving for the **Order_No_Out** output parameter.

The **Update Order** component opens and pauses at the breakpoint at the first step. It seems that the problem was solved by the new step in the **Create Order** component, but to confirm, click in the margin next to the **OK Button** step to add a breakpoint. This will pause the component run immediately after the component inserts the order number retrieved from the previous component.



Click the **Run** button. When the component pauses, note that the number inserted in the Open Order number dialog box of the Flight Reservations application matches the number that you saw in the **Order No** text box at the end of the previous component.

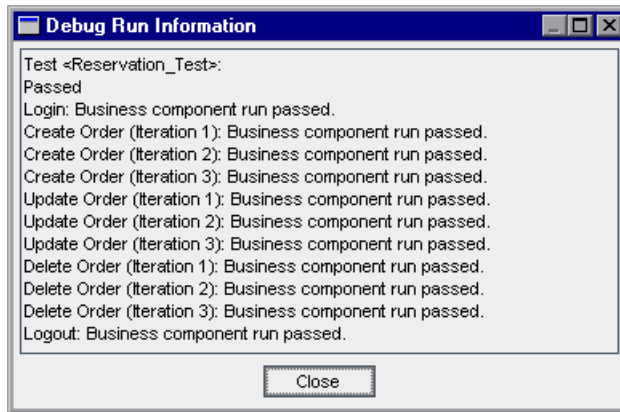
Click **Run** again to proceed. When the component ends it again pauses at the breakpoint at the first step of the Update Component. For the second and third iterations of this component, you can click **Run** to let the component complete its run.

The **Delete Order** and **Logout** components are set to run in **Normal** mode. After the three **Update Order** component runs, the **Delete Order** component runs three times and immediately following, the **Logout** component runs once and closes the Flight Reservation dialog box.

8 View the test results.

When the run is complete, the last component in the test (Logout) remains open in QuickTest. Display Quality Center by clicking the Microsoft Internet Explorer icon in the task bar.

Review the summary information of the test and the component iterations in the Debug Run Information dialog box.



Note the order of the iteration runs for the **Create Order**, **Update Order** and **Delete Order** components: 1-2-3, 1-2-3, 1-2-3.

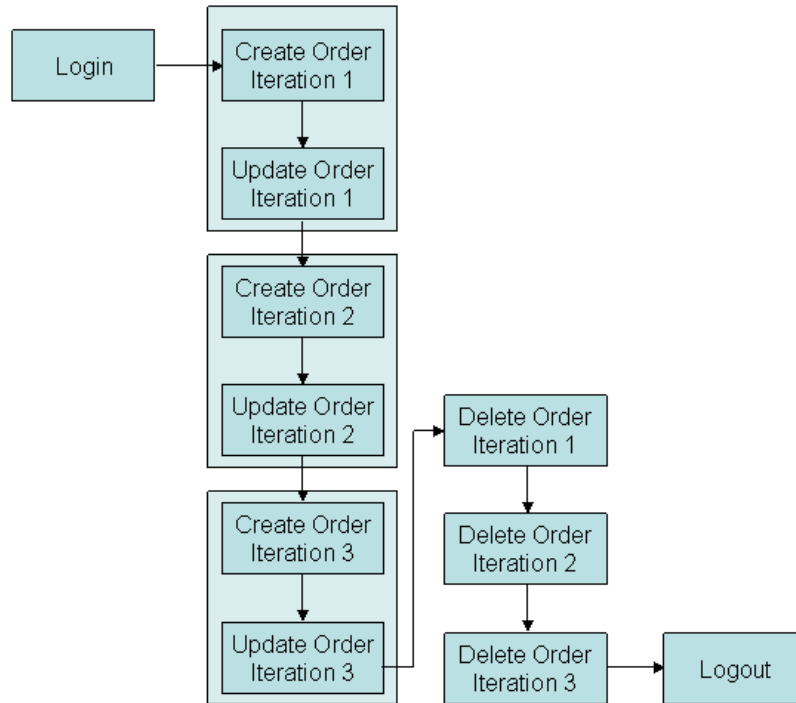
Click **Close** to complete the test run.

Now that you have successfully debugged your test, you are ready to add finishing touches to your test by grouping components and selecting iterations to run.

Grouping Components

In this section, you combine the **Create Order** and **Update Order** components so that they iterate as a group.

The process is illustrated as follows:



After opening the Flight Reservations application and logging in, the **Create Order** component creates an order to the first destination city, and the **Update Order** component updates the class. New orders are then created and updated for the second and third destinations. The business process test then deletes these orders (to return the flight reservation database to its previous status) and closes the application.

1 Display the components in the test.

Ensure that the **Reservation_Test** business process test is selected in the test plan tree, and the **Test Script** tab is displayed.

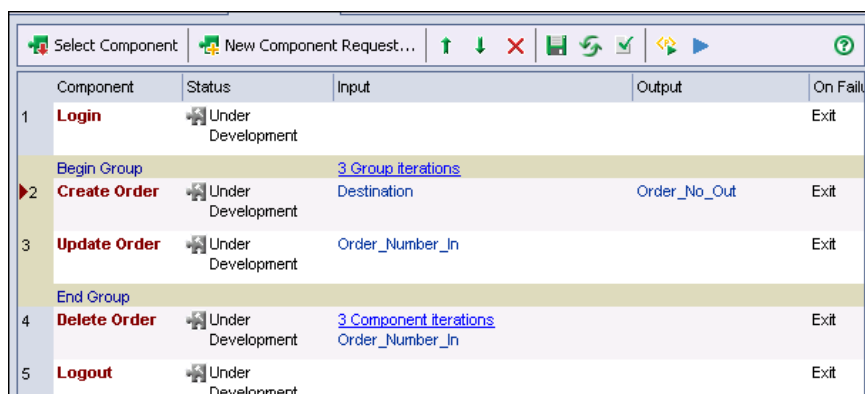
2 Group selected components.

In the Test Script tab, select both the **Create Order** and **Update Order** business components.

You select multiple components using standard Windows selection techniques (using the left mouse button and SHIFT key).

Click to the left of the **Create Order** component row. Ensure that the entire row is selected (the whole row should be highlighted). Hold down the SHIFT key and select the **Update Order** component. Both component rows are highlighted.

Right-click and choose **Grouping > Group Components** from the context menu. The group is created as indicated by the border around the selected components.



| Component | Status | Input | Output | On Fail |
|-----------------------|-------------------|-----------------|--------------|---------|
| 1 Login | Under Development | | | Exit |
| Begin Group | | | | |
| 2 Create Order | Under Development | Destination | Order_No_Out | Exit |
| 3 Update Order | Under Development | Order_Number_In | | Exit |
| End Group | | | | |
| 4 Delete Order | Under Development | Order_Number_In | | Exit |
| 5 Logout | Under Development | | | Exit |

3 View the iteration settings.

Click the group iterations link in the group border to open the Group Iterations dialog box and view the iteration settings. Note that both the **Destination** and **Order_Number_In** parameters are displayed in the same dialog box and they retain the configuration settings you assigned in the previous sections of this lesson. Click **OK** to close the dialog box.

4 Save the changes.

Click the **Save** button on the tab toolbar.

5 Run the components to confirm that the test runs successfully.

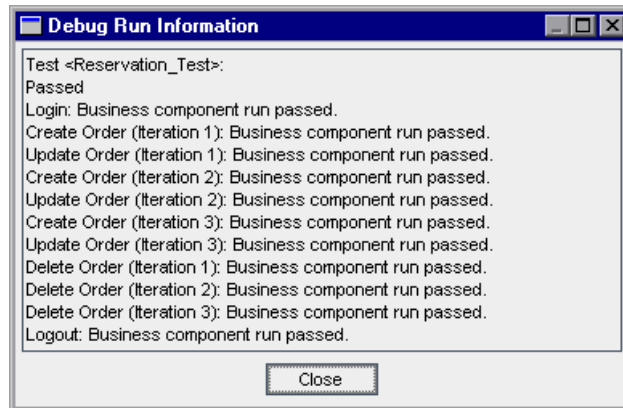
Click the **Run or Debug Test** button in the toolbar.

The Run or Debug Test dialog box opens. Click the **Set All to Normal** button so that all components will run in **Normal** mode. Click **OK** to close the Run or Debug Test dialog box and run the business process test.

6 View the test results.

When the run is complete, the last component in the test (Logout) remains open in QuickTest. Display Quality Center by clicking the Microsoft Internet Explorer icon in the task bar.

Review the summary information of the test and the component iterations in the Debug Run Information dialog box.



Note that the order of the iteration runs for the **Create Order**, **Update Order** and **Delete Order** components differs from the previous run without the grouping: 1-1, 2-2, 3-3, 1-2-3.

Click **Close** to complete the test run.

Now that you have created and debugged your business process test, you can proceed to Lesson 8, “Running Business Process Tests” In Lesson 8, you will learn how to select iterations to run and how to create run-time input parameter values, set test iterations, run business process tests in test sets, and view test results in the Test Lab module of Quality Center.

8

Running Business Process Tests

This lesson teaches you how to select which component iterations to run, how to create run-time input parameter values, and how to set test iterations.

Run-time parameters provide values to business components when the business process test is run. This enables you to iterate entire tests using variable input values that can provide differing test results.

The lesson describes how to view detailed reports showing the results of the business process test when the run is complete. In contrast with the simple pass or fail-type results provided with debug runs, these results include details on the configuration of each business component, the actual value of each component parameter used in the test, and the results of individual steps. This enables you to identify failed steps, or determine whether the results of your test were as expected.

In this lesson you will learn about:

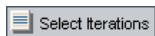
- Selecting Iterations to Run
- Creating a Run-Time Parameter
- Setting Up a Business Process Test in the Test Lab Module
- Running the Business Process Test in the Test Lab Module
- Viewing Run Results

Note: This lesson focusses specifically on setting up and running business process tests in test sets in the Test Lab module. For general information about working with the Test Lab module, refer to the *Mercury Quality Center User's Guide*.

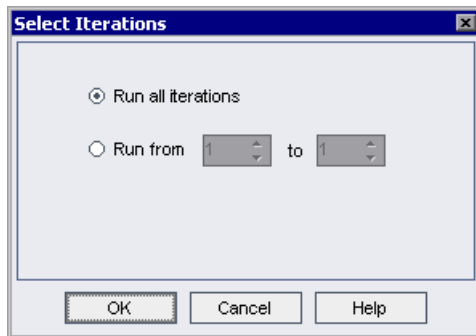
Selecting Iterations to Run

After you have configured parameter values for a component or a group of components, you can select to run only one or a specific range of iterations. In this section you define that only two of the three destination parameters you defined are input during the run (and as a result, only two order numbers are retrieved in the subsequent components).

1 Select iterations to run.



In the Test Script tab, click the Iterations link for the group in the **Input** column. The Group Iterations dialog box opens. Click the **Select Iterations** button in the toolbar. The Select Iterations dialog box opens.



Select the **Run from** radio button. The **from** and **to** spin boxes are enabled.

Click the up arrow in the **to** box to change the iteration number to **2**. This specifies that only iterations 1 and 2 will run in the business process test.

Click **OK**.

The selected range of component iterations (Selected for running 1-2) is displayed in the top left cell of the Group Iterations dialog box, and the iteration that is not included in the range (Iteration #3) is grayed in the iteration list.

Group Iterations

Add Iteration
 Delete Iteration
 Select Iterations
 Import...
 Export...

| | Create Order | Update Order |
|----------------------------|---|--|
| Selected for running 1 - 2 | Destination (string) <input type="checkbox"/> Output of previous component | Order_Number_In (number) <input checked="" type="checkbox"/> Output of previous component |
| Iteration #1 | Paris | [Create Order] Order_No_Out (Iteration 1) |
| Iteration #2 | Seattle | [Create Order] Order_No_Out (Iteration 2) |
| Iteration #3 | Zurich | [Create Order] Order_No_Out |
| | | |

Parameter <[Create Order] Destination> description:

Destination cities. Possible values: Denver, Frankfurt, London, Los Angeles, Paris, Portland, San Francisco, Seattle, Sydney, Zurich.

Click **OK** in the Group Iterations dialog box. The selected range is displayed in the Design Steps tab together with the Iterations link.

| | Component | Status | Input | Output | On Failure |
|---|---|-------------------|------------------------|--------------|------------|
| 1 | Login | Under Development | | | Exit |
| | 3 Group iterations (Selected for running 1 - 2) | | | | |
| 2 | Create Order | Under Development | Destination | Order_No_Out | Exit |
| 3 | Update Order | Under Development | Order_Number_In | | Exit |
| | 3 Group iterations (Selected for running 1 - 2) | | | | |
| 4 | Delete Order | Under Development | 3 Component iterations | | Exit |
| | | Iterations error | Order_Number_In | | |
| 5 | Logout | Under Development | | | Exit |

2 Fix the iterations error.

When you selected the iterations to run for the group, you instructed the test to create only two orders. However, the test is currently set to delete three orders. You must change the Delete Order iteration settings to match the group iteration settings.

Click the **Delete Order** iterations link. The Component Iterations dialog box opens.



Click the **Select Iterations** button in the toolbar. The Select Iterations dialog box opens.

Change the iteration settings. The total number of iterations must match for components whose parameters refer to previous components, but the specific iteration numbers you choose can be different. Select the **Run from** radio button. Click the up arrow in the **from** box to change the iteration number to **2**. Click the up arrow in the **to** box to change the iteration number to **3**. This specifies that only iterations 2 and 3 will run in the business process test.

Click **OK** to close the Select Iterations dialog box. Click **OK** to close the Component Iterations dialog box.

3 Save the changes.



Click the **Save** button on the tab toolbar.

4 Run the components.



Click the **Run or Debug Test** button in the toolbar.

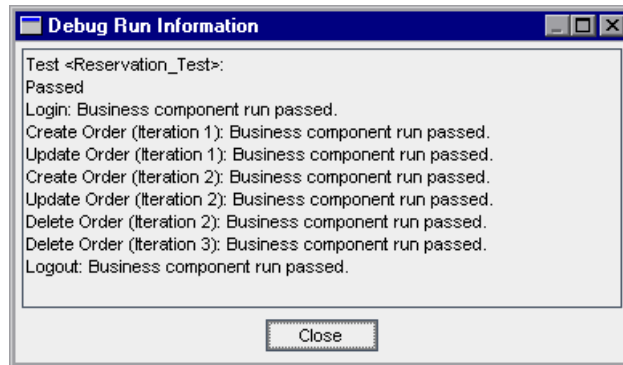
The Run or Debug Test dialog box opens.

Click **OK** to close the Run or Debug Test dialog box and run the business process test.

5 View the Results.

When the run is complete, display Quality Center by clicking the relevant button on the task bar.

Review the summary information of the test and the component iterations in the Debug Run Information dialog box.



Click **Close** to complete the test run.

Creating a Run-Time Parameter

In this section, you define a run-time parameter for the Agent Name step in the **Login** component. You will later provide two input values for this run-time parameter in order to run two consecutive iterations of your business process test, each with a different agent name.

1 Create a new parameter for the Login component.

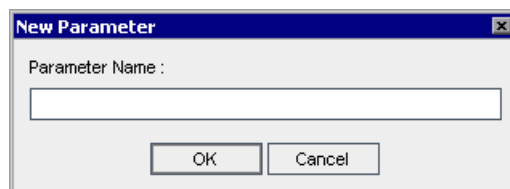


Click the **Business Components** module button in the sidebar. The Business Components module opens.

Select the **Login** business component in the component tree, and click the **Parameters** tab.



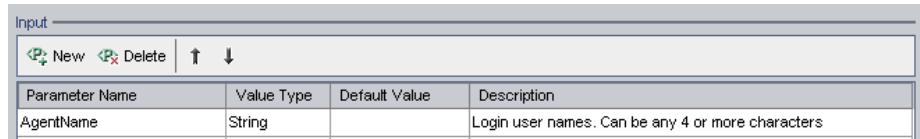
In the Input pane toolbar, click the **New Input Parameter** button. The New Parameter dialog box opens:



Enter **AgentName** in the **Parameter Name** box and click **OK**.

A line for the new input component parameter is added in the Input parameter list in the Parameters tab. **String** is entered automatically in the **Value Type** cell.

Leave the **Default Value** cell blank. In the **Description** cell, enter a description of the component parameter's purpose, or other descriptive text. For example: Login user names. Can be any 4 or more characters.



| Parameter Name | Value Type | Default Value | Description |
|----------------|------------|---------------|---|
| AgentName | String | | Login user names. Can be any 4 or more characters |

2 Change the Agent name step from a fixed value to a component parameter.

Click the **Automation** tab.



In the **Agent Name** row, click in the **Value** column. The **Configure the value** button is displayed. Click the button. The Value Configuration Options dialog box opens.



Click the **Parameter** radio button and ensure that **Component parameter** is selected. Click the down arrow next to the parameter name to display the description of the parameter you created in the previous step.

Click **OK**. The Value Configuration Options dialog box closes.

3 Save the changes to your component.



Click the **Save** button in the Automation tab toolbar.

4 Define the parameter as a run-time parameter.



Click the **Test Plan** module button in the sidebar to open the Test Plan module.

Click the **Test Script** tab for the **Reservation_Test** business process test. The new input parameter you created, and a link, are displayed in the **Input** column of the **Login** component.

Input Parameter Link

| Component | Status | Input | Output | On Failure |
|-----------------------|-------------------|--|--------------|------------|
| 1 Login | Under Development | AgentName: [Link] | | Exit |
| Begin Group | | | | |
| 2 Create Order | Under Development | 3 Group iterations (Selected for running 1 - 2) Destination | Order_No_Out | Exit |
| 3 Update Order | Under Development | Order_Number_In | | Exit |
| End Group | | | | |
| 4 Delete Order | Under Development | 3 Component iterations (Selected for running 2 - 3) Order_Number_In | | Exit |
| 5 Logout | Under Development | | | Exit |

The input parameter link is blank because no value has been defined for it yet. Click the input parameter link in the **Input** column. The Component Iterations dialog box opens.

Click in the **Parameter Value** cell, click the list arrow, and select **<New Run-Time Parameter>** from the list. **{Enter param name}** is displayed in the cell.

Overwrite the **Enter param name** text between the { } brackets with a descriptive name for the run-time parameter, for example, **AgentName**.

Component <Login> Iterations

Specify the parameter values that the business component will use for each iteration during a business process test run. Alternatively, select the Output parameter check box to use an output parameter value from a previous business component as input for this component.

Add Iteration
 Delete Iteration
 Select Iterations
 Import...
 Export...

| | |
|-----------------|---|
| | AgentName (string) <input type="checkbox"/> Output of previous component |
| Parameter Value | {AgentName} |

Parameter <AgentName> description:
Login user names. Can be any 4 or more characters

OK Cancel Help

Make sure that the `{ }` brackets are not deleted, as they instruct Quality Center to treat the component parameter as a run-time parameter and not as a fixed value.

5 Close the Component Iterations dialog box.

Click **OK**. The new run-time parameter name for the **Login** component is displayed in the Test Script tab between `{ }` brackets.

Setting Up a Business Process Test in the Test Lab Module

In the Test Lab module, you create test sets and choose which business process tests to include in each set. In this lesson, you create the test set that contains the single business process test you created in Test Plan. You then define iterations for the test.

Setting up a business process test requires the following steps:

- Creating a Test Set in a Folder
- Adding a Test to your Test Set
- Defining Iterations for the Business Process Test

Creating a Test Set in a Folder

In this section, you create a new folder containing a test set.

1 Open Test Lab and display the test sets tree.



Click the **Test Lab** module button in the sidebar to open the Test Lab module.

If the Test Lab tree view is not already displayed, choose **View > Test Sets Tree**.

2 Define a new test folder.

In the test sets tree, select the **Root** folder.



Click the **New Folder** button in the toolbar. The New Folder dialog box opens.

Enter BPT_Tutorial and click **OK**. The new folder appears under **Root** in the test sets tree.

3 Add a test set to your folder.



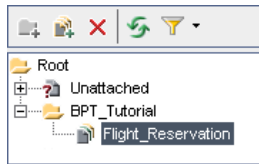
Ensure the new folder is selected in the test sets tree, and click the **New Test Set** button in the toolbar. The New Test Set dialog box opens.

In the **Test Set Name** box, enter Flight_Reservation.

In the **Description** box, enter a description of the test set, for example, This test set contains only the Reservation_Test business process test.

Click **OK**.

Flight_Reservation is added to the **BPT_Tutorial** test set folder in the test sets tree.

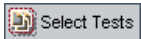


Adding a Test to your Test Set

Once you have defined a test set, you can add your business process test to the test set.

1 Display the test plan tree.

Select the **Flight_Reservation** test set in the test sets tree.



If the Test Plan tab is not currently displayed on the right side of the screen, click the **Select Tests** button in the Execution Grid tab toolbar. The right pane displays the test plan tree from the Test Plan module.

2 Add your test to the test set.

In the test plan tree, expand the **BPT_Tutorial** folder and select **Reservation_Test**.

Double-click or drag to add the test to the Execution Grid.



Click the **Close** button to close the Test Plan Tree pane.

Defining Iterations for the Business Process Test

In this section, you configure the business process test to run iterations according to the run-time parameter values that you define.

- 1 Select the **Flight_Reservation** test set in the test sets tree. The business process test that comprises the test set is displayed in the Execution Grid tab.

| Execution Grid Execution Flow Test Set Properties Linked Defects | | | | | | |
|--|---------------------|------------------|--------|------------|--------------------|-----------|
| | Plan: Test Name | Plan: Type | Status | Iterations | Responsible Tester | Exec Date |
| | [1]Reservation_Test | BUSINESS-PROCESS | No Run | Undefined | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

- 2 Right-click **Reservation_Test** in the tab and choose **Iterations**. The Test Iterations dialog box opens.

| Test <[1]Reservation_Test> Iterations | |
|---|--|
| Add Iteration Delete Iteration Select Iterations Import... Export... | |
| AgentName (string) | |
| Iteration #1 | |
| Parameter description: | |
| <input type="text"/> | |
| <input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/> | |



In the Test Iterations dialog box, click the **Add Iteration** button. An additional iteration row is added.

In the Iteration cells, enter a name for each agent, for example **AgentOne**, and **AgentTwo**. The business process test will log into the Flight Reservations application twice, using a different login name in each iteration of the test.

Click **OK** to close the Test Iterations dialog box.

Running the Business Process Test in the Test Lab Module

You can now run the business process test.

1 Reset the Exit conditions for all components.

In Lesson 7, “Working with Parameters,” you temporarily set all components to exit on failure while you debugged your test. Now that your test runs smoothly, you are ready to run your test to test new builds of your application. You can reset the exit conditions so that they only exit the test run if it does not make sense to continue the run after such a failure.



Click the **Test Plan** module button in the sidebar to open the Test Plan module, and click the **Test Script** tab, if it is not currently displayed. Select **Continue** for the **Update Order**, **Delete Order**, and **Logout** components. Since the **Update Order** and **Delete Order** components are dependant on the success of the **Login** and **Create Order** components, keep the **Exit** condition for these components.

2 Save the changes.

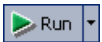


Click the **Save** button on the tab toolbar.

3 Define the test execution details.

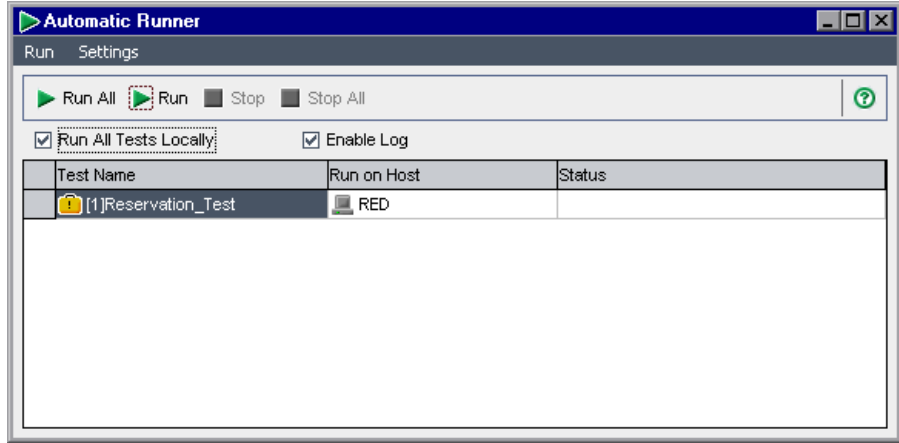


Click the **Test Lab** module button in the sidebar to open the Test Lab module.



Select the business process test in the Execution Grid and click the **Run** button in the tab toolbar.

The Automatic Runner dialog box opens.



Note: When you click the **Run** button, the Automatic Runner dialog box lists the specific business process test selected in the Execution Grid. When you click the **Run Test Set** button, the dialog box lists all the tests in the test set.

Ensure that the **Run All Tests Locally** check box is selected. This runs the business process test using QuickTest Professional on your local computer. (If you want to run the test on a different host computer, select the cell in the **Run on Host** column, and enter a valid host name that has QuickTest Professional 9.0 installed.)

Ensure that the **Enable Log** check box is selected. This enables QuickTest to record an execution log of the test run.

4 Run the test.



Select the **Reservation_Test** business process test in the dialog box and click the **Run** button in the toolbar.

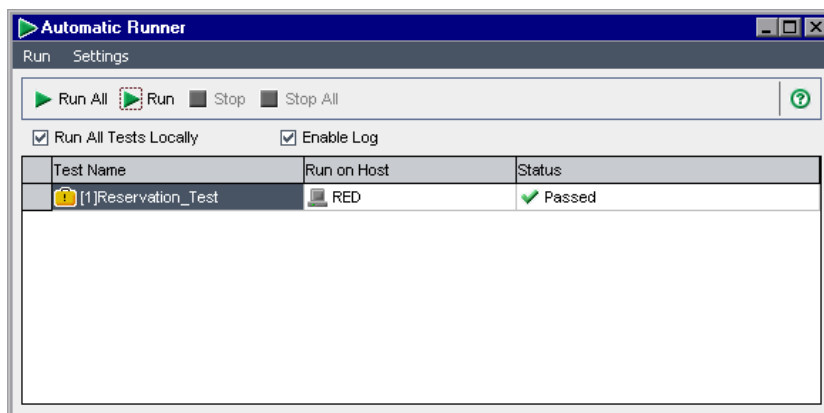
QuickTest Professional opens and runs the business process test in the test set.

From your task bar, you can toggle between the Test Lab module, QuickTest, and the Flight Reservations application.

- The status of the test run, for example, **Connecting**, or **Running**, is shown in the **Status** column of the Automatic Runner dialog box.
- The status of the test run is shown in QuickTest Professional in the status bar at the bottom of the QuickTest window.

After the **Logout** component closes the Mercury Flight Reservations application and the test run ends, display the Test Lab module in Quality Center.

The overall result of the run is shown in the **Status** column of the Automatic Runner dialog box.



After you run a business process test from the Test Lab module, you can view detailed results, as described in the next section.

Viewing Run Results

You can view the results of your business process test in various formats. This section describes the following:

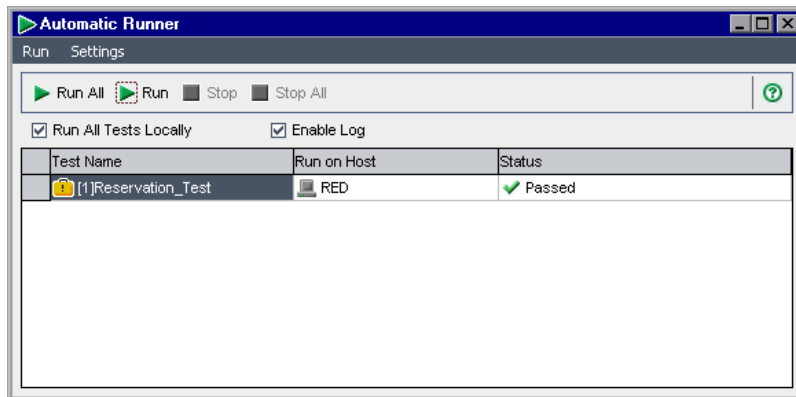
- Viewing the Execution Log
- Viewing Last Run Results in the Test Lab Module
- Viewing the QuickTest Report

Viewing the Execution Log

The Execution Log provides a log of the business process test (or tests) run in a test set. The log includes the date and time that each test was run, the host on which it was run, and the results of the tests.

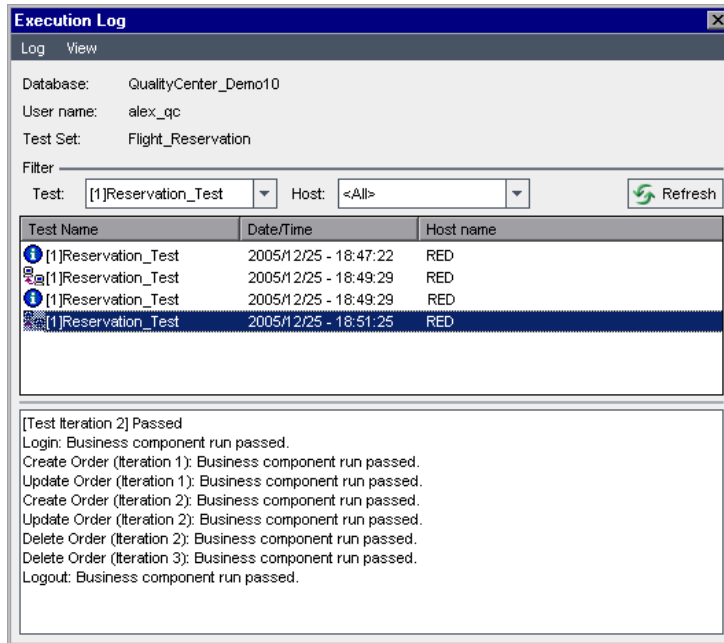
1 Open the Execution Log.

After the test run is complete, the Automatic Runner dialog box displays the Pass or Fail status of the test run.



Double-click in the dialog box.

The Execution Log window opens and displays the results of the test run.



The box in the middle of the window lists the different operations in the business process test, such as the connection to the host, or the results of each test iteration.

Select each line in turn. The results are displayed in the box at the bottom of the Execution Log window.

2 Close the Execution dialog box.



In the Execution Log dialog box, choose **Log > Exit** or click the **Close** button to return to the Execution dialog box.



In the Execution dialog box, choose **Run > Exit** or click the **Close** button to return to the Test Lab window.

Viewing Last Run Results in the Test Lab Module

You view the results of the latest business process test run in the Test Lab module in the Last Run Result pane at the bottom of the Execution Grid. The pane displays a hierarchical tree of each business component and step contained in the business process test, and the result obtained when the step ran, that is, **Passed**, or **Failed**. The run date and time is also shown.

Note: If one or more steps within a component receive a **Failed** result, the component itself will display a **Failed** status.

1 Display the results of the last test run.



Ensure that the Last Run Result pane is displayed at the bottom of the Execution Grid tab. If the Last Run Result pane is not visible, click the **Show** button at the bottom-right of the Execution Grid tab.

When the Last Run Result pane is first accessed, the tree displays only the iterations in the business process test.

2 View the run results.

Click the **Expand** symbol to the left of each iteration to view the result of each component. Then expand each component to view the main events in that component.

| Last Run Result | | | |
|----------------------------|--------|------------|-----------------------|
| Name | Status | Exec Date | Description: |
| Test Iteration 1 | Passed | 12/25/2005 | |
| Test Iteration 2 | Passed | 12/25/2005 | |
| Login | Passed | 12/25/2005 | Expected: |
| Create Order (Iteration 1) | Passed | 12/25/2005 | Actual: |
| Update Order (Iteration 1) | Passed | 12/25/2005 | Parameters: |
| Create Order (Iteration 2) | Passed | 12/25/2005 | Input |
| Update Order (Iteration 2) | Passed | 12/25/2005 | Destination = Seattle |
| Delete Order (Iteration 2) | Passed | 12/25/2005 | Output |
| Delete Order (Iteration 3) | Passed | 12/25/2005 | Order_No_Out = 42 |
| Logout | Passed | 12/25/2005 | |

Select each of the items in the pane in turn to view a description of relevant information in the **Description** area.

Viewing the QuickTest Report

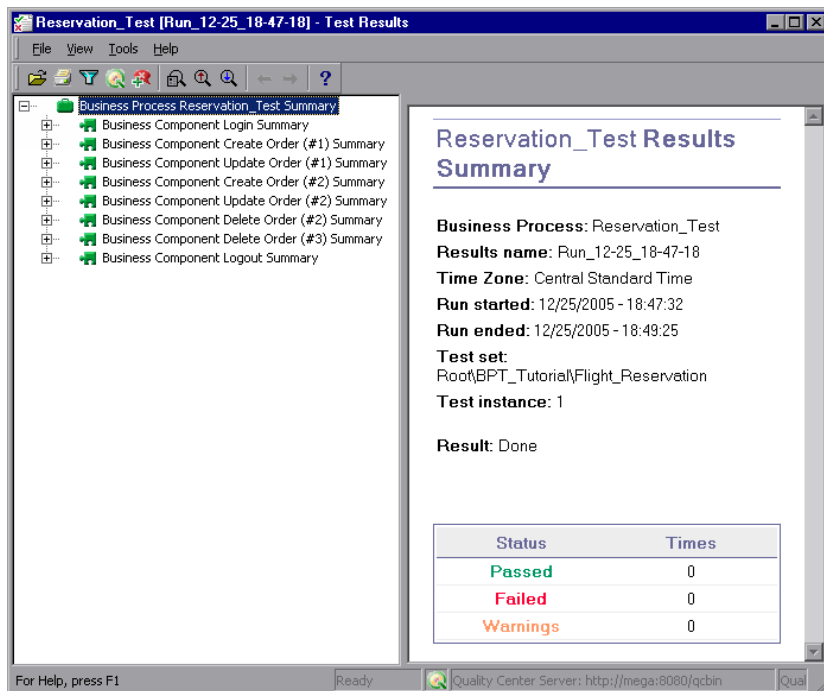
You can view a full QuickTest report of a business process test run from the Test Lab module. This hierarchical report contains details on all the steps for all the different iterations and business components within the test run.

1 Open the Test Results window.

View the **Last Run Result** pane at the bottom of the Execution Grid tab, as described in “Viewing Last Run Results in the Test Lab Module” on page 109.

Click the **Passed** link in the **Status** column next to **Test Iteration 1**.

The QuickTest Test Results window opens.



The Test Results window displays a graphical representation of the result of each step in the business process test.

2 View detailed results for selected steps.

In the Test Results menu, choose **View > Expand All** and then select nodes in the tree to view the details of how the steps performed in the test.

3 View the results for Test Iteration 2.

Close the QuickTest Test Results window for Test Iteration 1. In the **Last Run Result** pane, click the **Passed** link in the **Status** column next to **Test Iteration 2**. The QuickTest Test Results window for Test Iteration 2 opens.

Expand the nodes in the tree to view the details of how the steps performed in the test.

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

9

Where Do You Go from Here?

Now that you have completed the lessons in this tutorial, you are ready to apply the Business Process Testing concepts and skills you have learned to your applications.

In this lesson, you will learn about:

- Getting Started
- Getting Additional Information
- Documentation Updates

Getting Started

In this tutorial, you learned the basic workflow for creating automated keyword-driven components and business process tests in Quality Center as a Subject Matter Expert.

Setting up Application Areas

The tutorial provided you with a pre-made object repository and other resources in a sample application area.

Good application areas are one of the main keys to successful keyword-driven business process testing. To test your own applications, you should work closely with an Automation Engineer to set up one or more application areas that fit the needs of your applications and testing requirements. You should coordinate with your expert to ensure that the object names in the object repository and the function names in the library files are names that will be easy for you to identify when you select items and operations from the Quality Center keyword view.

Learning About Other Features

This tutorial touched on many of the mainstream Business Process Testing features, but before you go forward with creating tests on your own applications, you may want to familiarize yourself with additional features, including:

- Running business process tests manually
- Adding snapshots to business components
- Viewing component Used-by lists
- Component and test statuses
- Validating business process tests
- Using the Document Generator to generate information about your business components and business process tests
- Working with WinRunner components
- And more....

Mercury provides many resources to help you learn about these and other Quality Center and Business Process Testing features, as described in the next section.

Getting Additional Information

Online Resources



Quality Center includes the following online resources:

Readme provides last-minute news and information about Quality Center and Business Process Testing.

What's New describes the newest features in the latest versions of Quality Center and Business Process Testing.

Books Online displays the complete documentation set in PDF format. Online books can be read and printed using Adobe Reader which can be downloaded from the Adobe Web site (<http://www.adobe.com>).

Quality Center Online Help provides immediate answers to questions that arise as you work with Quality Center. It describes menu commands and dialog boxes, and shows you how to perform Quality Center and Business Process Testing tasks.

Customer Support Online uses your default Web browser to open the Mercury Interactive 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. The URL for this Web site is <http://support.mercury.com>.

Mercury Interactive on the Web uses your default Web browser to open Mercury's home page. This site provides 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. The URL for this Web site is <http://www.mercury.com>.

Documentation Updates

Mercury is continuously 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** In the Customer Support Web site, click the **Documentation** link.
- 2** Under **Please Select Product**, select Mercury Quality Center.

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

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

