# HP Unified Functional Testing

For the Windows ® operating systems

Software Version: 12.01

GUI Testing Tutorial

## Legal Notices

### Warranty

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

The information contained herein is subject to change without notice.

### Restricted Rights Legend

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

### Copyright Notice

© Copyright 1992 - 2014 Hewlett-Packard Development Company, L.P.

### Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Google™ and Google Maps™ are trademarks of Google Inc

Intel® and Pentium® are trademarks of Intel Corporation in the U.S. and other countries.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

## Documentation Updates

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

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

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **http://h20230.www2.hp.com/selfsolve/manuals**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

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

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

## Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

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

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

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

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

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

To find more information about access levels, go to:

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

**HP Software Solutions Now** accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

# Contents

# Welcome to the UFT Tutorial for GUI Testing

The UFT Tutorial for GUI Testing is a self-paced guide that teaches you the basics of testing your GUI application with UFT. It will familiarize you with the process of creating and running automated GUI tests and analyzing the run results.

After completing the tutorial, you can apply the skills you have learned to testing your own application or Web site.

> **Note:** To learn more about creating and running API tests, see the UFT Tutorial for API Testing, available from the `< UFT installation folder>\help` directory.

## Tutorial Audience and Scope

This tutorial is intended for users who are new to UFT. No prior knowledge of UFT or QuickTest is required. A general understanding of testing concepts and functional testing processes may be helpful, but is not mandatory. UFT enables you to create GUI tests, business process tests, and API tests. This tutorial reviews topics related to GUI testing only. While performing the lessons in this tutorial, you may notice unfamiliar menu items or other UFT GUI elements that are not described in the tutorial. These may be relevant for BPT or API testing only, and are not relevant for GUI testing at all. For details about these elements, see the *HP Unified Functional Testing User Guide*.

> **Note:** This tutorial refers to file system paths that are relevant for Windows 7 operating systems. The paths in other operating systems may be slightly different.

## Using UFT with BPT

In addition to tests, UFT enables you to create service and keyword-driven business components for use in business process tests, if you are connected to an ALM server that supports BPT. The procedures described in this tutorial are designed for creating GUI tests, but you can also apply the majority of these procedures to creating keyword-driven components. For more details on business components and BPT, see the *HP Unified Functional Testing User Guide* and the *HP Business Process Testing User Guide*.

> **Note:** Unless otherwise specified, references to **Application Lifecycle Management** or **ALM** in this guide apply to all currently supported versions of ALM and Quality Center. Note that some features and options may not be supported in the specific edition of ALM or Quality Center that you are using.
>
> For a list of the supported versions of ALM or Quality Center, see the *HP Unified Functional Testing Product Availability Matrix*, available from the UFT help folder or the HP Support Matrix page (requires an HP passport).
>
> For details on ALM or Quality Center editions, see the *HP Application Lifecycle Management User Guide* or the *HP Quality Center User Guide*.

# UFT Guides and References

The following tables provide a list of the UFT guides, online help and references:

**Note:** To check for recent updates of any of the guides below, visit the HP Software Product Manuals Web site (http://h20230.www2.hp.com/selfsolve/manuals).

**Getting started**

| Reference | Description |
|---|---|
| What's New? | Describes the newest features in the latest version of Unified Functional Testing.<br><br>You can also access the **What's New** from the **Unified Functional Testing Help** menu. |
| Product Movies | Click the link or select **Help > Product Feature Movies** to view short movies that demonstrate the main product features. |
| Readme | Provides last-minute news and information about Unified Functional Testing.<br><br>For the latest readme file, go to the HP Software Manuals Web site (requires an HP Passport) at http://support.openview.hp.com/selfsolve/manuals. |
| UFT PAM | The Product Availability Matrix (PAM) provides current information about technologies and integrations supported for this version of UFT. |
| GUI Testing Tutorial | The GUI Testing Tutorial is a self-paced printable guide, designed to lead you through the process of creating GUI tests and familiarize you with the testing environment. |
| API Testing Tutorial | The API Testing Tutorial is a self-paced printable guide, designed to lead you through the process of creating API tests in the Windows environment. |

**PDF guides**

| Guide | Description |
|---|---|
| UFT User Guide | The HP Unified Functional Testing User Guide describes how to use UFT to test your applications. It provides step-by-step instructions to help you create, debug, and run tests, and report defects detected during the testing process. |
| Run Results Viewer | The HP Run Results Viewer User Guide explains how to use the Run Results Viewer to interpret and use the test results from your GUI or API tests. |
| UFT Installation Guide | The HP Unified Functional Testing Installation Guide provides a complete, step-by-step instructions on how to install and set up UFT on a standalone computer. |

| Guide | Description |
|---|---|
| UFT QuickStart | The UFT Installation QuickStart Sheet explains the steps to perform a basic installation of UFT. |
| License Server Installation Guide | The Concurrent License Server Installation Guide provides the information you need to install and maintain the HP Functional Testing Concurrent License Server. |
| UFT Add-ins Guide | The HP Unified Functional Testing Add-ins Guide explains how to set up support for UFT add-ins and standard Windows testing support. Add-ins enable you to test any supported environment using GUI tests and business components. |
| What's New | Describes the newest features in the latest version of Unified Functional Testing. |
| Security Reference | Contains security relevant information for using UFT and references for further information |

**References**

Links to the references are available from the UFT online help home page.

| Reference | Description |
|---|---|
| Object Model Reference | The Object Model Reference for GUI Testing includes a description, a list of methods and properties, syntax, examples, and identification properties for each UFT test object. |
| VBScript Reference | Microsoft's Visual Basic Scripting documentation that describes objects, methods, properties, functions, and other elements that can be used when writing VBScript scripts. |
| Automation Object Model Reference | List the objects, methods, and properties that enable you to control UFT from within another application. |
| Object Repository Automation Reference | Describes the objects that enable you to manipulate UFT shared object repositories and their contents from outside of UFT. |
| Run Results Schema Reference | Provides details about the structure of the Run Results XML schema, and describes the elements and attributes used in the its XML reports. |
| Test Object Schema Reference | A reference describing the elements and attributes available for creating test object configuration XML content, for use when creating UFT extensibility projects. |

| Reference | Description |
|---|---|
| Object Repository Schema Reference | Describes the elements and complex types defined for the object repository schema. |

For additional online references, see the Welcome section of a PDF guide.

# Additional Online Resources

The following additional online resources are available:

| Resource | Description |
|---|---|
| **HP Software Support Online** | The HP Software Support Web site (www.hp.com/go/hpsoftwaresupport). To access, this choose **Help > HP Software Support**.<br><br>**Note:** Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport user ID, go to: http://h20229.www2.hp.com/passport-registration.html |
| **Testing Forums** | • GUI Testing: http://h30499.www3.hp.com/t5/Unified-Functional-Testing/bd-p/sws-Fun_TEST_SF<br><br>• API Testing: http://h30499.www3.hp.com/t5/Service-Test-Support-and-News/bd-p/sws-Serv_TEST_SF<br><br>• BPT: http://h30499.www3.hp.com/t5/Business-Process-Validation/bd-p/sws-BPT_SF |
| **UFT Product Page** | The HP Unified Functional Testing product page (http://www8.hp.com/us/en/software-solutions/unified-functional-testing-automated-testing/index.html), with information and related links about UFT. To access, select **Help > Useful Links > Product Page**. |
| **Troubleshooting & Knowledge Base** | The Troubleshooting page (http://h20230.www2.hp.com/troubleshooting.jsp) on the HP Software Support Web site where you can search the HP Software Self-solve knowledge base. To access, select **Help > Knowledge Base** or **Help > Troubleshooting**. |
| **HP Software Community** | The HP IT Experts Community site (http://h10124.www1.hp.com/campaigns/IT_Experts/pages/home.html), where you can interact with other HP software users, read articles and blogs on HP software and access downloads of other software products. |

| Resource | Description |
|---|---|
| **HP Manuals Site** | The HP Software Product Manuals Web site (http://support.openview.hp.com/selfsolve/manuals), to search for the most up-to-date documentation for a selected HP Software product. To access, select **Help > Useful Links > HP Manuals Site**. |
| **What's New** | The UFT What's New Help, describing the new features and enhancements in this version of UFT. |
| **Product Movies** | The UFT HPLN (HP Live Networks) page (https://hpln.hp.com/page/uft-120-videos) with a list of all product movies. |
| **HP Software Web site** | The HP Software Web site (www.hp.com/go/software). This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. |

You can access the following sample applications from the **Start** menu. These applications are the basis for many examples in this guide:

- Mercury Tours sample Web site. The URL for this Web site is http://newtours.demoaut.com.

- Mercury Flight application. To access from the Start menu, select **All Programs > HP Software > HP Unified Functional Testing > Sample Applications > Flight API / Flight GUI**.

  These applications are also available from the file system at: `<UFT installation folder\samples\flight\app\flight4a.exe` (for the Flight GUI application) or `<UFT installation folder>\samples\flight_service\HPFlights_Service.exe` (for the Flight API application)

# Chapter 1: Introducing Unified Functional Testing - GUI Testing

This lesson provides you with an overview of automated testing and UFT, and gets you set up to perform the tutorial.

**Note:** It is recommended to work with an soft copy of this tutorial because there are sections in which you will be asked to copy and paste test steps into UFT. However, keep in mind that in selected parts of this tutorial, UFT will be recording your operations or learning objects that you point to with the mouse. During such sessions, you should refrain from switching focus to the PDF or Help window.

You can open a PDF of this tutorial from your **<Unified Functional Testing installation folder>\help folder** or by selecting **Help > Printer-Friendly Documentation** in UFT.

You can access the HTML version of this tutorial, by selecting **Help > Unified Functional Testing Tutorial**.

This chapter includes:

# Benefits of Automated GUI Testing

If you have ever tested applications or Web sites manually, you are aware of the drawbacks of manual testing.

Manual testing is time-consuming and tedious, and requires a heavy investment in human resources. Worst of all, time constraints often make it impossible to manually test every feature thoroughly before the application is released. This leaves you wondering whether serious bugs have gone undetected.

Automated GUI testing with UFT addresses these problems by dramatically speeding up the testing process. You can create tests that check all aspects of your application or Web site, and then run these tests every time your site or application changes.

As UFT runs tests, it simulates a human user by moving the cursor in a Web page or application window, clicking GUI objects, and entering keyboard input. However, UFT does all this faster than any human user.

| Benefits of Automated Testing | |
|---|---|
| Fast | Automated tests are significantly faster than human users. |
| Reliable | Tests perform precisely the same operations each time they are run, thereby eliminating human error. |
| Repeatable | You can test how the Web site or application reacts after repeated execution of the same operations. |
| Programmable | You can program sophisticated tests that bring out hidden information. |
| Comprehensive | You can build a suite of tests that covers every feature in your Web site or application. |
| Reusable | You can reuse tests on different versions of a Web site or application, even if the user interface changes. |

# Testing Process

The UFT testing process consists of the following main phases:

1. **Analyzing your application.**

   The first step in planning your test is to analyze your application to determine your testing needs.

   - **What are your application's development environments?** You will need to load UFT add-ins for these environments to enable UFT to identify and work with the objects in your application. Examples of development environments include Web, Java, and .NET.

   - **What business processes and functionality do you want to test?** To answer this, think about the various activities that customers perform in your application to accomplish specific tasks.

   - **How can you break your test into small testable units and tasks?** You should break the processes and functionality you want to test into smaller tasks, so that you can create UFT actions based on those tasks. Smaller and more modular actions make your tests easier to read and follow, and help ease maintenance in the long run.

   At this stage, you can already begin creating test skeletons and adding actions.

2. **Preparing the testing infrastructure.**

   Based on your testing needs, you must determine what resources are required and create these resources accordingly.

   Examples of resources include **shared object repositories**, which contain test objects that represent objects in your application, and **function libraries**, which contain functions that enhance UFT functionality.

   You also need to configure UFT settings so that UFT will perform any additional tasks you may you need, such as displaying a results report every time you run a test.

3. **Building your tests and adding steps to each test.**

   After the testing infrastructure is ready, you can begin building your tests.

   You can create one or more empty tests and add actions to them to create the testing skeletons. You associate your object repositories with the relevant actions, and associate your function libraries with the relevant tests, so that you can insert steps using keywords.

   You can also add all your tests to a single solution. A solution enables you to store, manage, and edit any related tests together, without having to close one test before opening another.

   You may also need to configure test preferences at this point.

4. **Enhancing your test.**

   - You can test whether your application is functioning correctly by inserting **checkpoints** into your tests. Checkpoints search for a specific value of a page, object, or text string.

   - You can broaden the scope of your test and test how your application performs the same operations with multiple sets of data. Do this by replacing fixed values with **parameters**.

   - You can add other sophisticated checks to your test by adding **programming and conditional or loop statements** and other programming logic to your test using VBScript.

     This tutorial includes steps for checkpoints and parameters only. For additional details about VBScript statements, see the *HP Unified Functional Testing User Guide*.

5. **Debugging, running, and analyzing your test.**

   You can debug your test using debugging functionality to ensure that it operates smoothly and without interruption. After the test is working correctly, you run it to check the behavior of your application. While running, UFT opens the application and performs each step in your test.

   Examine the run results to pinpoint defects in your application.

6. **Reporting defects.**

   If you have ALM installed, you can report the defects you discover to a database. ALM is the HP test management solution.

   For details about working with ALM, see the *HP Application Lifecycle Management User Guide*.

# UFT Window

Before you begin creating tests, familiarize yourself with the main UFT window.

The image below shows a UFT window as it would appear right after you create a test, with the test flow shown in the canvas, and with the toolbar, Solution Explorer, Data pane, and Properties pane displayed:
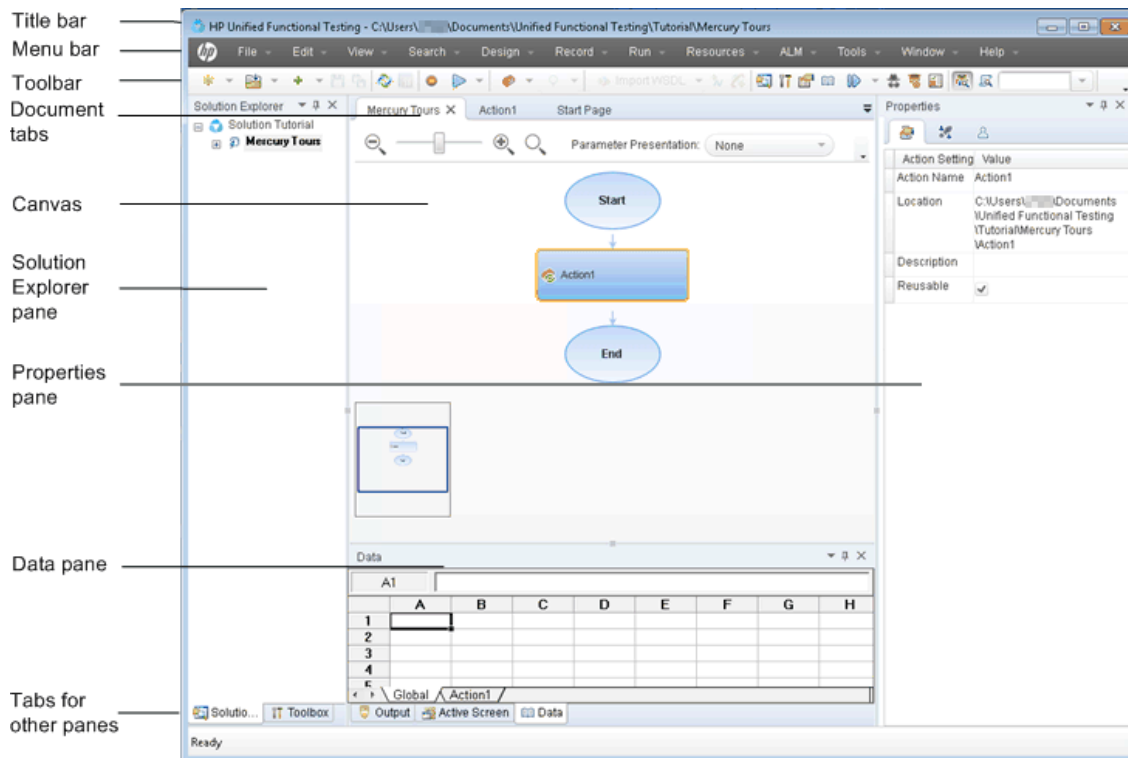


Take a few minutes to explore the components of the main UFT window. You can then continue to analyze the application you will be testing in this tutorial, as described in "Analyzing Your Application and Creating Actions" on page 20.

## Document Types

UFT displays open documents in the document pane. Use the document tabs located just below the toolbar to navigate to open documents and bring them into focus.

The document pane can display the following types of files:

- **Tests.** You can create, view, and modify your test in the Canvas, which enables you to edit the flow of your test, action, or component.

- **Actions or Business Components**. Enables you to create, view, and modify your action or business component, in the following views:

  - **Keyword View**. Contains each step, and displays the object hierarchy, in a modular, icon-based table.

- ■ **Editor**. Contains each step as a VBScript line, displayed in a text and code editor. In object-based steps, the VBScript line defines the object hierarchy.

- **Function Library**. Enables you to create, view, and modify functions (operations) for use with your test. Function libraries open as separate tabs in the document pane, similar to tests or actions.

- **Application Areas**. Enables you to edit the application area settings and resource associations.

- **Start Page**. Welcomes you to UFT and provides links to recent files, descriptions of new features, product forums, and other support links. You can use the shortcut buttons to open new and existing documents.

- **Product Movies**. Accessible from the Start Page or **Help** menu, and teaches you about some of the main UFT features using descriptive and instructional movies.

- **Internal Browser Pages**. Enables you to open internet pages for forums and other product related page, such as those accessible from the Start Page or the **Help** menu.

## Toolbars and Menus

In addition to the document pane, the UFT window contains the following key elements:

- **Title bar.** Displays the path of the current test or solution.

- **Menu bar.** Displays menus of UFT commands.

- **UFT toolbar.** Contains commonly used buttons to assist you in designing your testing documents.

For details, see the *HP Unified Functional Testing User Guide*.

## Panes

Some of the panes and toolbar options listed below are described in detail in subsequent lessons. For details on other panes and toolbar options, see the *HP Unified Functional Testing User Guide*.

| Name | Toolbar Button | Description | Default Location |
|------|------|------|------|
| **Solution Explorer** | | Displays all the resources associated with your current test, and enables you to manage these resources. | A tab on the left side of the UFT window.<br><br>To display:<br><br>• Select **View > Solution Explorer**.<br><br>• Click the Solution Explorer button in the toolbar. |
| **Toolbox** | | Displays all the keywords available to your test, and enables you to drag and drop objects, or calls to functions, from the Toolbox pane into your test. | A tab on the left side of the UFT window.<br><br>To display:<br><br>• Select **View > Toolbox**.<br><br>• Click the **Toolbox** button in the toolbar. |
| **Document pane** | N/A | Displays all open documents. Each document has a tab that you can click to bring the document into focus. | An unlabeled pane in the center of the UFT window. Each document tab is labeled with the document name.<br><br>To display: Open a testing document. |
| **Properties** | | Displays all properties for the currently selected test, action, component, or application area. | A pane on the right side of the UFT window.<br><br>To display:<br><br>• Select **View > Properties**.<br><br>• Click the **Properties** button in the toolbar. |

| Name | Toolbar Button | Description | Default Location |
|------|----------------|-------------|------------------|
| **Data** |  | Assists you in parameterizing your test. | A tab at the bottom of the UFT window.<br><br>To display:<br><br>• Select **View > Data**.<br><br>• Click the **Data** button in the toolbar. |
| **Output** | N/A | Displays information sent during the run session. | A tab at the bottom of the UFT window.<br><br>To display: Select **View > Output**. |
| **Errors** | N/A | Displays a list of syntax errors found in your test or function library.<br><br>Provides a list of the resources that are referenced in your test but cannot be found, such as missing Recovery Scenarios and missing Function Libraries. | A tab at the bottom of the UFT window.<br><br>To display: Select **View > Errors**. |
| **Active Screen** | N/A | Provides a snapshot of your application as it appeared when you performed a certain step during a recording session. | A tab at the bottom of the UFT window.<br><br>To display: Select **View > Active Screen**. |
| **Debug panes** |  | Assists you in debugging your test.<br><br>The Debug panes include the **Breakpoints**, **Call Stack**, **Local Variables**, **Console**, and **Watch** panes. | Tabs at the bottom of the UFT window. Not displayed by default.<br><br>To display:<br><br>• Select **View > Debug**<br><br>• Click the **Debug** button in the toolbar, and then select the individual debug pane from the drop-down list. |

| Name | Toolbar Button | Description | Default Location |
|---|---|---|---|
| **Tasks** | N/A | Displays and enables you to manage the tasks defined for the current test.<br><br>Displays the TODO comment steps of the test's actions or currently open function libraries. | A tab at the bottom of the UFT window.<br><br>To display: Select **View > Tasks**. |
| **Search Results** | N/A | Displays all occurrences of the search criteria you define in the Find dialog box or using other Search menu items. | A tab at the bottom of the UFT window.<br><br>To display:<br><br>• Select **View > Search Results**.<br><br>• Perform a search. |
| **Bookmarks** | N/A | Displays the location of bookmarks in your action, scripted component, or function library, and enables you to navigate to these bookmarks. | A tab at the bottom of the UFT window.<br><br>To display: Select **View > Bookmarks**. |

# Accessing UFT in Windows 8 Operating Systems

By default, you can access UFT directly from the **Start** or **Apps** Screen in Windows 8.x.

In addition, you can add UFT application and files that were accessible from the **Start** menu in previous versions of Windows to the **Start** screen, including:

- **Applications (.exe files).** For example:

  - The Run Results Viewer

  - All UFT tools, such as the Password Encoder and the License Validation Utility

  - The API testing sample Flight applications

- **Non-program files.** You can access documentation and the link to the Mercury Tours Website from the **Apps** screen.

**Note:** By default, the Start and Apps screens on Windows 8 are set to open Internet Explorer in Metro Mode. However, if User Account Control is turned off on your computer, Windows 8 will not open Internet Explorer in Metro mode.  Therefore, if you try to open an HTML shortcut from

the Start or Apps screen, such as the UFT Help or Readme file, an error will be displayed.

To solve this, you can change the default behavior of Internet Explorer so that it never opens in Metro mode. In the **Internet Properties** dialog box > **Programs** tab, select **Always in Internet Explorer on the desktop** for the in the **Choose how you open links** option. For more details, see http://support.microsoft.com/kb/2736601 and http://blogs.msdn.com/b/ie/archive/2012/03/26/launch-options-for-internet-explorer-10-on-windows-8.aspx.

# Chapter 2: Analyzing Your Application and Creating Actions

"Introducing Unified Functional Testing - GUI Testing" on page 10 gave you an overview of automated testing and UFT, and helped you set up your system to perform the tutorial.

In this lesson, you will analyze an application to determine what needs to be tested, and you will learn to create an action in UFT. An action is a logical unit in your test that focuses on testing a particular part of your application.

This chapter includes:

# Getting to Know Your Application

Before you begin creating tests, you must determine exactly what you want to test in your application. To do this, you need to analyze your application in terms of its business processes— the distinct activities that customers perform in your application to complete particular tasks.

For the purpose of this tutorial, you need to become familiar with the Mercury Tours Web site. Mercury Tours is a sample web application that simulates a web-based flight information and reservation service. You can access the Mercury Tours Web site using any web browser, but you can perform the tutorial using only Microsoft Internet Explorer.

As you navigate the site and familiarize yourself with its content, consider the following:

- **How is the application organized?** Are there separate areas for each activity? What are these areas? What are these activities? What are the expected outcomes for each activity?

- **What screens must be tested in each of these areas?** What objects need to be tested in each screen?

- **What activities might be used in multiple scenarios?** One example of such an activity is logging on to the application. Think of these activities as "reusable" parts.

  Analyzing your application in terms of its separate parts can help you to design your tests so that they comprise small, modular units, which are easier to understand, debug, and maintain over time.

- **What business processes might users perform using this application?** One scenario discussed in this tutorial is reserving flights. Other scenarios might include making cruise or hotel reservations, or even registering on the Web site.

- **Which development environments need to be supported for testing purposes?** UFT provides add-ins to support many testing environments. You load these add-ins when you open UFT. When working with this tutorial, you need to load the Web Add-in because Mercury Tours is a Web application. Your application might require you to load Java or other add-ins.

# Explore the Mercury Tours Web Site

Open the Mercury Tours Web site and explore it for a while. As you navigate the site and familiarize yourself with its content, consider the questions posed in "Getting to Know Your Application" above.

1. **Launch the Mercury Tours Web site.**

   In Microsoft Internet Explorer, browse to the Mercury Tours Web site:http://newtours.demoaut.com

   The Mercury Tours home page opens.

2. **Log in to Mercury Tours.**

In the Find a Flight area, enter `tutorial` in both the **User Name** and **Password** boxes and click **Sign-In**.

> **Note:** If you are prompted to have Internet Explorer remember the password, click **No**.

The Flight Finder page opens.



3. **Explore the Mercury Tours site.**

   Beginning with the Flight Finder page, follow the on-screen instructions to get flight information and reserve a flight.

As you navigate the application, think about what you might want to test, and which pages and objects need to be included in each test that you might create.

> **Note:** On the Book a Flight page, you do not have to fill out actual personal or credit card information; just enter the required information, indicated in red, in the **Passengers** and **Credit Card** sections. You may enter fictitious information.

4. **End your Mercury Tours session and close the browser.**

   a. When you finish exploring the Mercury Tours Web site, click **LOG OUT** on the Flight Confirmation page or **SIGN-OFF** at the top of any application page.

      To sign on again, simply enter `tutorial` in the **User Name** and **Password** boxes in the SIGN-ON page or in the Mercury Tours home page.

   b. After logging out, close the Internet Explorer window.

You are now ready to use this tutorial to create tests on the Mercury Tours Web site using UFT. See "Creating Tests" below for details, and then continue with "Create a New Solution, Test, and Actions" on the next page.

# Creating Tests

Each UFT test includes calls to **actions**. Actions are units that divide your test into logical sections. When you create a new test, it contains a call to one action. By dividing your tests into calls to multiple actions, you can design more modular and efficient tests.

You can also group your tests according to functionality and store them together in **solutions**. By adding tests and other documents to solutions, you can open and edit the documents in the same UFT session without having to open, save, and close each test as you modify it, and you can share resources across related tests.

## Internal and External Actions

UFT lets you insert several types of actions into tests. An **internal** action is an action that is stored in the local test (also known as a source test). An **external** action is a referenced call to an action that is stored in a different test. For example, you may have an action that you want to use in multiple tests—you would store the action as an internal action in one test and insert calls to that action from other tests. In the other tests the action would be available as an external action.

## Reusable Actions

When you insert a call to a new action, it is **reusable** by default, enabling you to insert calls to the action from any test.

When working with reusable actions, you only need to update the existing action stored with the original test. When you modify the original action, all tests containing calls to the reusable action are updated. Reusable actions are read-only in the calling test, and you modify them in the test in which they are stored.

Reusable actions are especially useful when:

- You have a process that you may need to include in several tests, such as logging in to your application

- You may have a process that you need to insert several times in the same test, such as filling in user credential dialog boxes that open whenever a user tries to access a secure part of your application.

If you want to prevent an action from being reused in any test, you can make it **non-reusable**. Non-reusable actions are internal actions that can be used only once.

> **Tip:** If you plan to use the same or similar actions in many tests, you might consider creating a repository test to store your reusable actions. Then you can insert calls to these existing actions from other tests, linking these actions to the source test. You can store the repository test in the same solution as your other tests, enabling easier access to your reusable actions.

### Copied Actions

You can also insert a copy of an action in your tests if you plan to make changes to the action steps. This inserts the action as an **internal** action in the current test. Copies of actions are not linked to the source test, so that when you insert a call to a copy of an existing action, you can make changes to the copied action, and your changes will neither affect, nor be affected by, any other test.

# Create a New Solution, Test, and Actions

In this exercise, you will create a new solution and test for the Mercury Tours Web site that you explored in "Explore the Mercury Tours Web Site" on page 21.

The actions you create in this exercise will be reused in subsequent lessons.

1. **Start UFT.**

   Do one of the following:

   - **If UFT is not currently open**:

     - Double-click the UFT icon  on your desktop or on the **Start** Screen (in Windows 8.x )

     - In Windows 7, select **Start > All Programs > HP Software > HP Unified Functional Testing**.

     > **Note:** For details on accessing UFT and UFT tools and files in Windows 8, see "Accessing UFT in Windows 8 Operating Systems" on page 18.

In the Add-in Manager, confirm that the **Web** Add-in is selected, and clear all other add-ins. Click **OK** to close the Add-in Manager and open UFT.

The UFT splash screen is displayed while UFT loads your selected add-ins. This may take a few seconds.

- **If UFT is already open**: Check which add-ins are loaded by selecting **Help > About HP Unified Functional Testing**. The add-ins are listed under **Installed Features**.

If the Web Add-in is not loaded, you must exit and restart UFT. When the Add-in Manager opens, select the Web Add-in, and clear all other add-ins.

> **Note:** If the Add-in Manager does not open when starting UFT, select **Tools > Options**. In the **Startup Options** pane of the **General** tab, select **Display Add-in Manager on startup**. When you exit and restart UFT, the Add-in Manager opens.

2. **Explore the Start Page.**

The Start Page provides links to recent files, information about new features in this version of UFT, as well as links to helpful support and community forums. In the top-right corner of the page, you can define options for displaying and closing the Start Page.

a. If it is not already displayed, in the Document Pane, click the **Start Page** tab.

b. In the top-right corner of the Start Page, select the **Options** drop-down arrow, and then select **Close Start Page after test loads**. The Start Page will close automatically after you create a test.

3. **Create a new test.**

a. Click the **New** button [ * ]. By default, UFT assumes that you want to create a new test, and the New Test dialog box opens.

b. Select **GUI Test**, populate the fields as follows:

   - **Name**: Enter MercuryTours.

   - **Location**: Tests are automatically saved at **C:\%HOMEPATH%\My Documents\Unified Functional Testing**, and you do not need to modify this path. An example of a default test location is **C:\Documents and Settings\<username>\My Documents\Unified Functional Testing**.

   - **Solution Name:** Enter Tutorial.

c. Select **Create directory for solution**. The Solution Location value is automatically updated with your new directory.
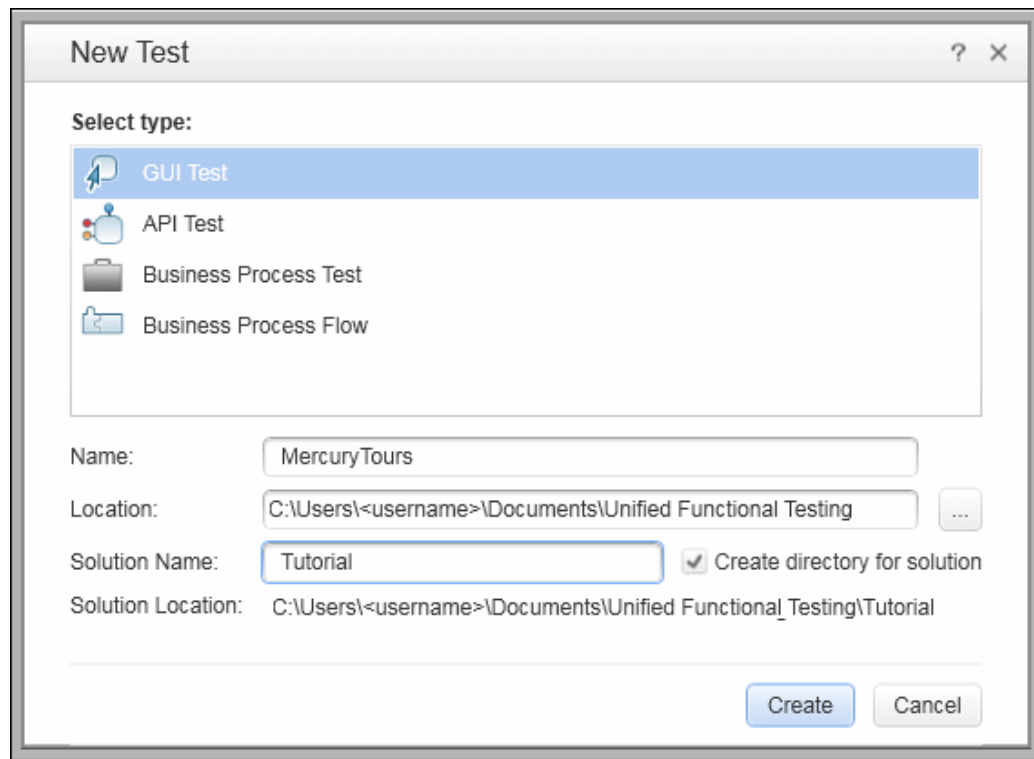
d. Click **Create.** A blank test opens, with the test flow displayed in the canvas, and a blank action tab opened in the background.

The test name (MercuryTours) and path are displayed in the title bar of the main UFT window.

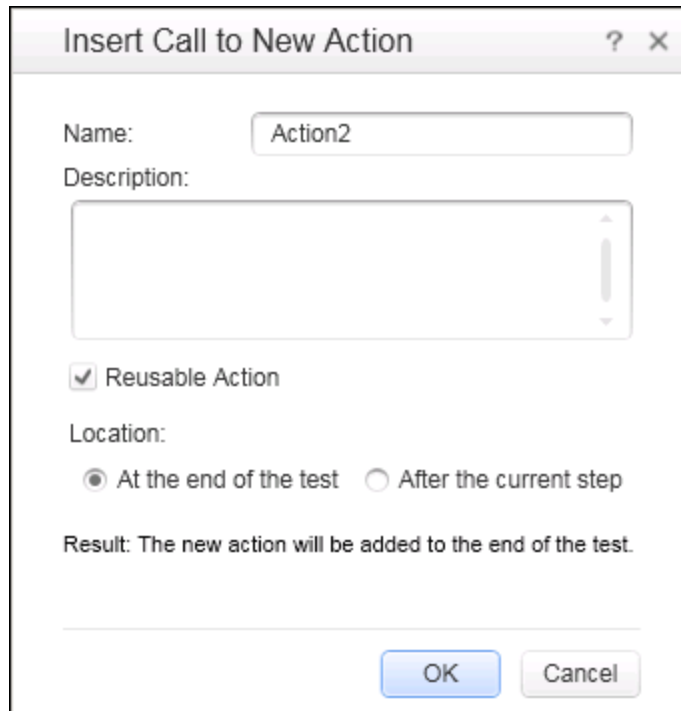In the Solution Explorer, you can see that the test is loaded as part of the Tutorial solution.

> **Note:** If the Solution Explorer is hidden, click the **Solution Explorer** button 📇 or select **View > Solution Explorer** to display it.

4. **Rename Action1 so that it has a logical name.**

   a. In the canvas, right-click the **Action1** image and select **Action Properties**.

   b. In the Action Properties dialog box, in the **Name** box, enter Login as the new name. In the message that opens, click **Yes**.

5. **Create a new action named FlightFinder.**

   a. Right-click anywhere in the canvas, and select **Call to New Action**.

   The Insert Call to New Action dialog box opens.



   b. In the **Name** box, enter `FlightFinder`, and confirm that both **Reusable Action** and **At the end of the test** are selected.

   c. Click **OK.** The dialog box closes and the **FlightFinder** action is added to the end of your test.

6. **Create calls to additional actions.**

   Add the following actions to your test:

   ▪ **SelectFlight**

   ▪ **BookFlight**

   Use the same steps you used in the previous step to create the FlightFinder action.

   Your test now contains all of the actions needed to begin this tutorial.

7. **Save your test.**

Click **Save** 🖫 .

> **Note:** You may have noticed that before you clicked save, an asterisk (*) was displayed in the MercuryTours document tab and Solution Explorer node. Asterisks are displayed to indicate that you have unsaved content, and disappear when you save the file. When you save a test, all test actions are saved too.

8. **Explore the test flow in the canvas.**

Click the **Mercury Tours** tab in the document pane to view the entire test flow in the canvas.

- You can zoom in or out on the test flow using the mouse or the zoom bar located above the canvas.

- Once the test flow is zoomed in to the maximum, and no longer fits in the canvas, you can also use the minimap in the bottom left corner of the canvas to navigate through the test flow.

If the minimap is not displayed, click the **Minimap** button above the canvas to display it. The minimap can be helpful if you have a test with many actions or nested actions.

You can continue with creating test objects for use in your test, as described in "Creating Object Repositories" on page 30.

## Types of Action Calls

You can insert the following types of action calls to your test:

- **Calls to new actions.** Inserts a new, empty action in your test.

- **Calls to a copy of an action.** Inserts a copy of an existing action in your test.

  The action is not linked to the source test or action from which you copied it.

- **Call to an existing action.** Inserts a call to an existing action in your test.

You can insert a call to an action that is stored in the current test (if you want to insert the same action twice, for example), or you can insert a call to an action from another test. The action remains linked to the source test from which you called it.

# Chapter 3: Creating Object Repositories

In "Analyzing Your Application and Creating Actions" on page 20 you learned about the Mercury Tours Web-based application and determined what needs to be tested. You then created a solution, test, and actions.

This lesson introduces test objects and run-time objects, and describes how to create shared object repositories that you can associate with the actions in your test.

This chapter includes:

# Introducing Test Objects and Object Repositories

Before you add steps to your test, you must set up the resources that will be used by that test. One of the most important resources for any test is the object repository.

An object repository is a storehouse for the test objects (and other object types) used in your test.

## Run-Time Objects

During a run session, UFT performs the operations on the run time objects in your application.

Test objects are stored representations of the actual run time objects (or controls) in your application. UFT creates test objects by learning a select set of the properties and values of the objects in your application, and uses that data to uniquely identify objects in your application during run time.

Each test object is part of a test object hierarchy. For example, a Link object may be part of a Browser/Page/Link hierarchy. Top-level objects, such as Browser objects, are known as container objects, as they can contain lower-level objects, such as Frame or Page objects.

In this lesson, you will use the Object Spy dialog box to view the properties and operations of objects in your application.

## Object Repositories

UFT can store the test objects it learns in the following types of object repository files:

- **Shared object repositories.** Contain test objects that can be used in multiple actions. This versatility makes it the preferred repository type for storing and maintaining test objects. By associating a shared object repository with an action, you make the test objects in that repository available for use in the action. Any maintenance you perform on an object in a shared object repository is reflected in all steps (within all actions) using that object.

- **Local object repositories.** Contain test objects that can be used in one specific action. The objects in this type of repository cannot be used in any other action. Local object repositories are useful for backing up your test objects, or for learning new objects.

During a run session, UFT references the test objects stored in the associated object repositories in order to perform operations on the corresponding run time objects in your application.

When you create an object repository, try to include only the objects you need for testing purposes. This keeps the object repository relatively small and helps to simplify maintenance and object selection. Also, make sure that you provide logical names so that others can easily select the correct objects when creating or modifying tests.

> **Note:** Object repositories can also include checkpoint and output objects. Checkpoint object types are covered in "Creating Checkpoints and Using Functions" on page 79.

# Learn Objects in Your Application

In this exercise, you will create an object repository for use in the Login action.

Use the UFT **Navigate and Learn** option to add objects from the Mercury Tours Web site to a shared object repository, and then associate the object repository with your test.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. On the Start page, in the Recent solutions area, click **Tutorial**.

   The Tutorial solution opens, displaying the MercuryTours test that you created in "Create a New Solution, Test, and Actions" on page 25.)

2. **Set the learn settings for UFT.**

   a. Select **Record > Record And Run Settings**. The Record and Run Settings dialog box opens.

   b. In the Web tab, select **Open the following address when a record or run session begins**.

   c. Confirm that the URL in the first box is http://newtours.demoaut.com. In the second box, select a browser on which to run your test session. Microsoft Internet Explorer is the browser used in this tutorial.

d.  Confirm that the following options are selected:

- **Do not record and run on browsers that are already open**

- **Close the browser when the test closes**
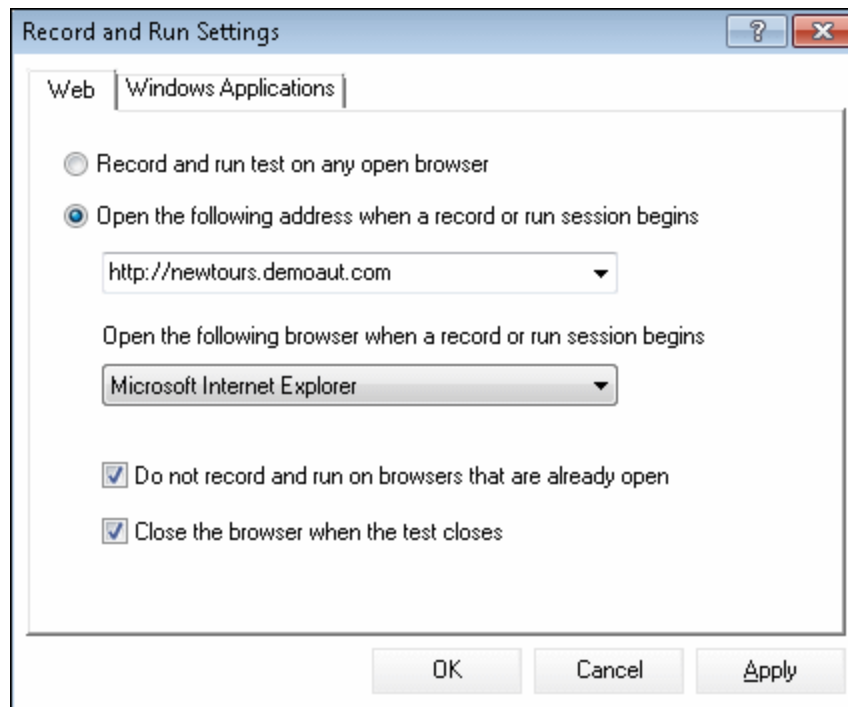


e.  Click **OK**. Later, when you run the test or record steps for it, your browser will open to the Mercury Tours Web site. When the record or run session ends, your browser will close.

3.  **Open the Mercury Tours Web site.**

Open Internet Explorer and browse to the Mercury Tours Web site: http://newtours.demoaut.com Ensure that the browser window is maximized.

> **Note:** Be sure to open a new Internet Explorer window *after*UFT. If you use a browser that was already open before you opened UFT, UFT may not properly recognize the objects in it.

4.  **View the properties and operations for some of the objects in the Mercury Tours Web site.**

a.  In UFT, click the **Object Spy** toolbar button . The Object Spy dialog box opens.

b.  Drag the Object Spy dialog box to the right side of your screen. This enables you to see the

objects you want to spy on more clearly. Verify that the **Keep Object Spy on top while spying** toggle button  is pressed.

c. Click the pointing hand button  .

UFT is hidden, and the Object Spy dialog box is displayed over the Mercury Tours Web site Welcome page.

> **Tip:** If you need to switch back and forth between Welcome page, UFT, or any other window currently open, press **CTRL** to change the pointing hand back to a regular Windows pointer. Hold down the **CTRL** button as long as you need the pointer, and then release it when you are ready to use the pointing hand.

d. Hover the mouse cursor over the various objects on the page and watch to see what happens in the Object Spy dialog box.

> **Note:** If UFT does not recognize your objects in the correct location, check to see that you are viewing the page at 100%, and are not zooming in or out of the page.
>
> For example, if you view the page at 90% or 120%, you may be required to click or select an area to the left or the right of the actual object in order to recognize it.

e. Position the mouse cursor in the **User Name** edit box (but do not click it). The test object hierarchy of that object is displayed. In the hierarchy box, notice that the name of the object is **WebEdit**, which is its object class type.

f. Click the **User Name** edit box. UFT opens again. In the Object hierarchy box, notice that the object class and name are now selected (**WebEdit:userName**).

g. Close the Object Spy dialog box.

5. **Start the Navigate and Learn process by setting up the Define Object Filter.**

In this step, you instruct UFT to learn only the objects that are needed for your test and add them to the object repository. This helps keep object repository maintenance to a minimum.

a. In UFT, select **Resources > Object Repository Manager**. The Object Repository Manager window opens.

b. In the Object Repository Manager window, select **Object > Navigate and Learn**. Both UFT and the Object Repository Manager are hidden, and the Navigate and Learn toolbar appears.
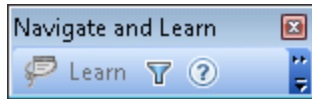


> **Note:** The first time you use Navigate and Learn, a message box displaying a tooltip opens. Click **OK** to enable the Navigate and Learn toolbar.

c. In the Navigate and Learn toolbar, click the **Define Object Filter** button  . The Define Object Filter dialog box opens.

d. In the Define Object Filter dialog box, select **Selected object types** and then click **Select**. The Select Object Types dialog box opens.

e.  Click **Clear All** to clear all of the check boxes. Then select the **EditBox** and **Image** check boxes and click **OK**.
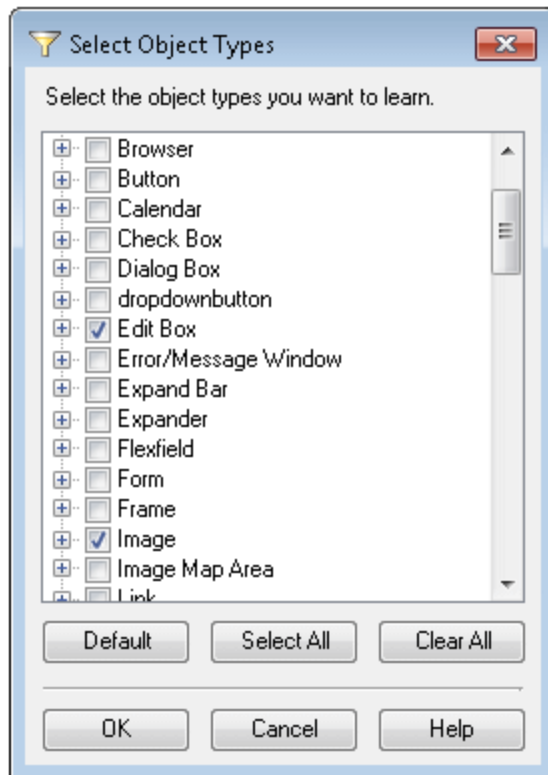


f.  In the Define Object Filter dialog box, click **OK**.

6.  **Learn the selected object types from the Welcome page.**

In this step, you instruct UFT to learn all of the objects in the Mercury Tours Web site Welcome page that match the filter you defined, and to add them to a shared object repository.

a.  In the Mercury Tours Web site Welcome page, click the **Welcome: Mercury Tours** title tab to bring it into focus as the Web page you want UFT to learn.

b.  In the Navigate and Learn toolbar, click **Learn**. The Web page flickers and the Adding Objects message box is displayed as UFT adds representations of the objects on the Web page to a new object repository.

> **Note:** Adding these objects takes a few seconds. Do not interact with the Web page while Navigate and Learn runs. Navigate and Learn has finished adding the objects to the object repository when the flashing stops and the Adding Objects box closes.
>
> Depending on the number of objects in the page, UFT can potentially take a longer time to learn all the objects.

    c.  Close the Navigate and Learn toolbar. UFT and the Object Repository Manager window are visible again.

7. **Remove extra objects from the object repository.**

In the previous step, UFT added all of the objects that match your filter selection to the object repository. You only need a few objects for this test, so in this step, you will delete the extraneous objects.

    a.  In the Object Repository Manager window, expand the objects in the **Test Objects** tree. You should see a list of objects similar to the following:
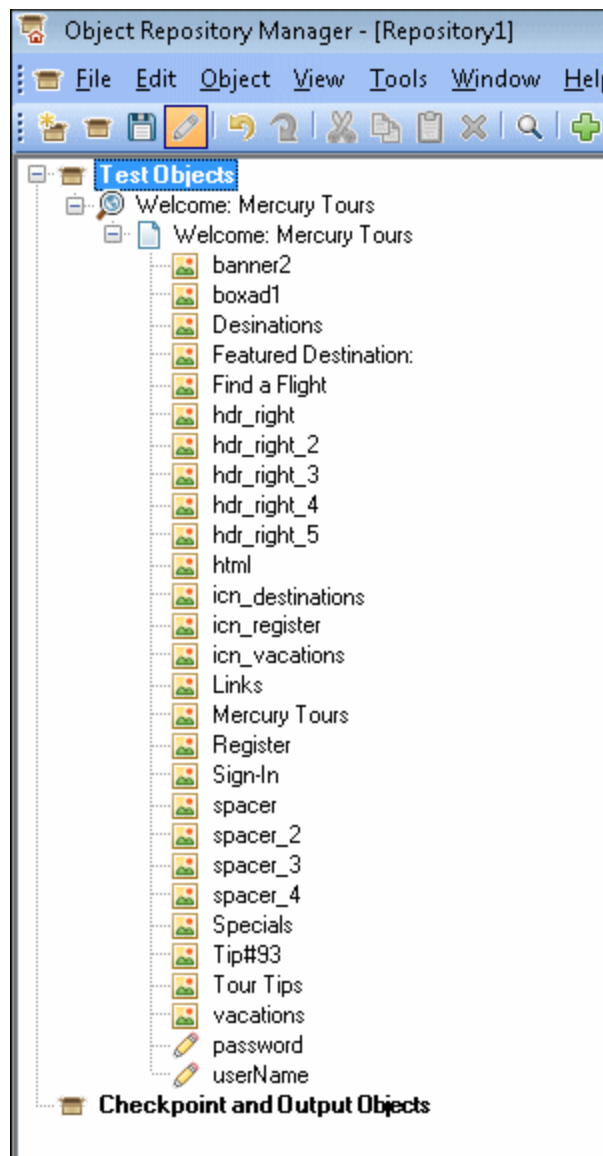
b. Delete all of the test objects *except* for **Sign-In** (image), **password** (edit box), and
   **userName** (edit box). Select the objects and press **Delete**. Click **Yes** in the confirmation
   box.

> **Note:** You can select multiple objects simultaneously using the SHIFT and CTRL keys.

After you delete the extraneous objects, your tree should resemble the following:
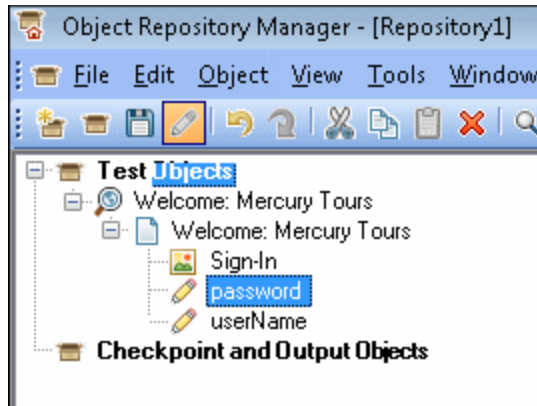


> **Tip:** If you delete **Sign-In**, **password**, or **userName** by mistake, click **Undo**  to
> restore the deleted objects.

8. **See what UFT learned about one of the objects.**

   In the **Test Objects** tree, select the **userName** object, and notice the object properties displayed in the right pane. These are the descriptive properties that UFT will use to identify the object during a run session.
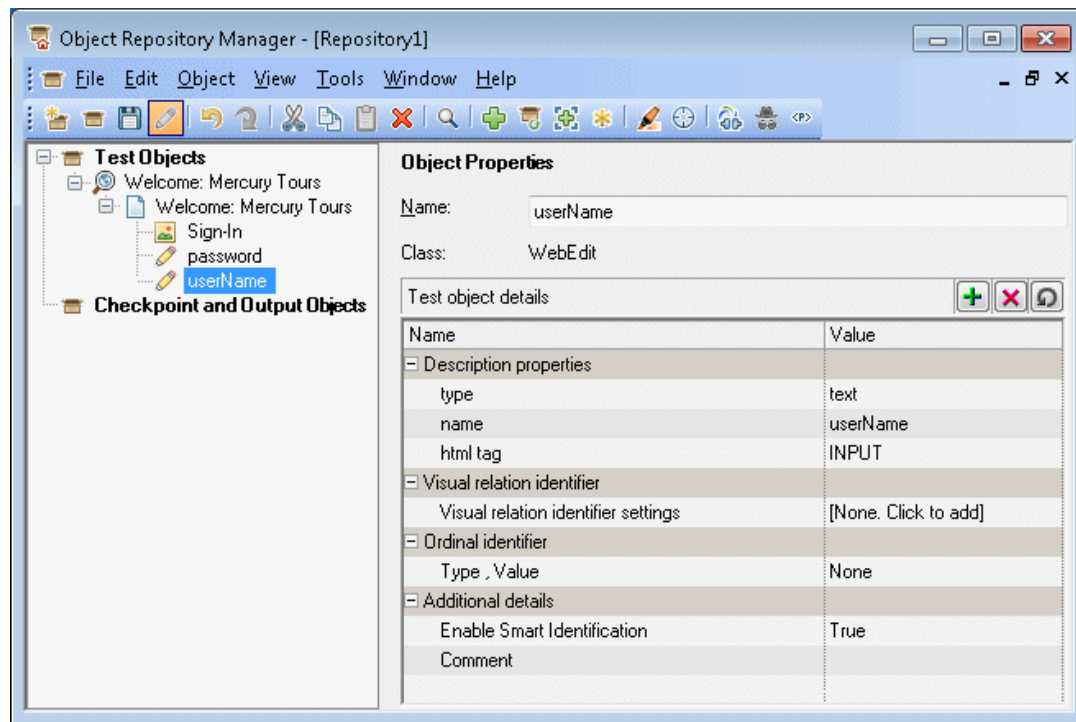


9. **Save the object repository.**

   a. In the Object Repository Manager window, click **Save** 💾 . The Save Shared Object Repository dialog box opens.

   b. Browse to the location in which you want to save your object repositories, such as **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**. Create a new folder named Tutorial_ObjectRepositories, and open it.   In the **File Name** box, enter MercuryToursLogin and click **OK**.

10. **Associate the object repository with the Login action.**

    Associating an object repository with an action enables you to use any object from that repository in any step in the associated action.

    a. Switch to the UFT window. If the Solution Explorer is not already open, open it by clicking the **Solution Explorer** button 🗔 .

    b. In the Solution Explorer, expand the **MercuryTours** test node.

    c.   Right-click the **Login** action and select **Associate Repository with Action**. The Open Shared Object Repository dialog box opens.

    d.   Click the **MercuryToursLogin.tsr** shared object repository and click **Open**.

    e.   In the dialog box that opens, click **Yes**.

       The object repository is now associated with the **Login** action, and is displayed in the Solution Explorer as a child of that action.

11.  **Save your test.**

    Select **File > Save <TestName>**.

Congratulations! You have created a shared object repository for the Welcome page of the Mercury Tours Web site, and associated it with your test's first action. Continue with creating object repositories for the other actions in your test and associating them with the relevant actions, as described in "Use Multiple Object Repositories" below.

# Use Multiple Object Repositories

In "Learn Objects in Your Application" on page 32, you created a shared object repository for the Welcome page of the Mercury Tours Web site. Now, you will create a shared object repository for each of the remaining pages in the site.

> **Tip:** It is recommended to always create a separate shared object repository for each page in your Web site or each area of your application. This makes it easier to find the correct object when adding or modifying test steps or when performing maintenance tasks.

For the purposes of this tutorial, you will instruct UFT to learn all of the objects on each page. At this point, you do not need to associate the object repositories with a specific action.

1.  **Log in to the Mercury Tours Web site Flight Finder page.**

    a.   If it is not currently open, open Internet Explorer and browse to the Mercury Tours Web site: http://newtours.demoaut.com.

    b.   In the **User Name** and **Password** boxes, enter **tutorial**.

    c.   Click **Sign-In**. The Flight Finder page opens.

2.  **Create a new shared object repository.**

    a.   If the Object Repository Manager is closed, open it by selecting **Resources > Object Repository Manager** in UFT. The Object Repository Manager window opens.

    b.  In the Object Repository Manager window, click **New** ![icon] . A blank shared object repository opens.

3.  **Start the Navigate and Learn process by setting up the Define Object Filter.**

    a.  In the Object Repository Manager window, select **Object > Navigate and Learn**. Both UFT and the Object Repository Manager are hidden.

    b.  In the Navigate and Learn toolbar, click the **Define Object Filter** button ![icon] . The Define Object Filter dialog box opens.

    c.  In the Define Object Filter dialog box, select **All object types** and click **OK**.
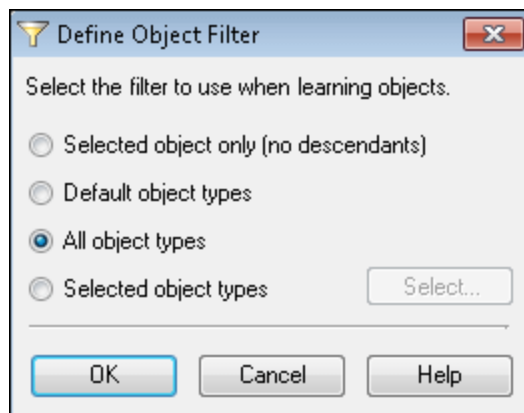


4.  **Learn all of the objects from the Flight Finder page.**

In this step, you instruct UFT to learn all of the objects in the Mercury Tours Web site Flight Finder page that match your filter, and to add them to a shared object repository.

    a.  In the Mercury Tours Web site Flight Finder page, click the **Flight Finder: Mercury Tours** title tab to bring it into focus as the Web page you want UFT to learn.

    b.  In the Navigate and Learn toolbar, click **Learn**. The Web page flickers and the Adding Objects message box is displayed as UFT begins adding representations of the objects on the Web page to a new object repository.

> **Note:** Adding these objects takes a few seconds. Do not interact with the Web page while Navigate and Learn runs.

    c.  Close the Navigate and Learn toolbar. UFT and the Object Repository Manager window are visible again.

5. **Save the shared object repository.**

   a. In the Object Repository Manager window, click **Save** 💾 . The Save Shared Object Repository dialog box opens.

   b. If you need to, browse to and open the **Tutorial_ObjectRepositories** folder. In the **File Name** box, enter `MercuryToursFlightFinder` and click **OK**.

6. **Create a new shared object repository for the Select Flight page.**

   a. In the Mercury Tours Web site Flight Finder page, click **Continue**. The Select Flight page opens. You are now ready to add objects from this page.

   b. Switch to the Object Repository Manager window and click **New** 📁 . A blank shared object repository opens.

   c. In the Object Repository Manager window, select **Object > Navigate and Learn**. Both UFT and the Object Repository Manager are hidden.

   d. In the Mercury Tours Web site Select Flight page, click the **Select Flight: Mercury Tours** title tab to bring it into focus as the Web page you want UFT to learn.

   e. In the Navigate and Learn toolbar, click **Learn**. The Web page flickers and the Adding Objects message box is displayed as UFT begins adding representations of the objects on the Web page to a new object repository.

   > **Note:** Adding these objects takes a few seconds. Do not interact with the Web page while Navigate and Learn runs.

   f. Close the Navigate and Learn toolbar. UFT and the Object Repository Manager window are visible again.

   g. In the Object Repository Manager window, click **Save** 💾 . The Save Shared Object Repository dialog box opens.

   h. If you need to, browse to and open the `Tutorial_ObjectRepositories` folder. In the **File Name** box, enter `MercuryToursSelectFlight` and click **Save**.

7. **Create object repositories for the remaining Web pages.**

   a. Using the process described in step 6, create new shared object repositories for each of the following pages:

      ○ **Book a Flight**

      ○ **Flight Confirmation**

b. Name the object repositories `MercuryToursBookFlight` and
`MercuryToursFlightConfirmation`, respectively.

8. **Associate the MercuryToursFlightFinder.tsr object repository with the FlightFinder action.**

a. Switch to the UFT window. If the Solution Explorer is not already open, open it by clicking

the **Solution Explorer** button .

b. In the Solution Explorer, right-click the **FlightFinder** action and select **Associate Repository with Action**. The Open Shared Object Repository dialog box opens.

c. Click the **MercuryToursFlightFinder.tsr** shared object repository and click **Open**.

d. In the dialog box that opens, click **Yes**

The object repository is now associated with the **FlightFinder** action, and is displayed in the Solution Explorer as a child of that action.

9. **Associate the remaining object repositories with the relevant actions.**

■ Associate the **SelectFlight** action with the **MercuryToursSelectFlight.tsr**.

■ Associate the **BookFlight** action with the following repositories:

○ **MercuryToursBookFlight.tsr**

○ **MercuryToursFlightConfirmation.tsr**

Later, when you add steps to each action, all of the required test objects will be available for use.

10. **Save your test.**

Click **Save**  .

Continue setting up your test infrastructure by adding additional resources to your test, as described in "Creating Functions and Function Libraries" on page 44.

# Chapter 4: Creating Functions and Function Libraries

UFT provides built-in functions and methods that will satisfy many of your testing needs. However, there may be times when you need to perform a task that is not available by default for a particular test object class. You can create a user-defined function to perform this task, save it in a function library file, and then insert it as a step every time that you need to perform this task.

In "Creating Object Repositories" on page 30, you created shared object repositories and associated them with the actions in your test. In this lesson, you will create a function and function library, and associate the function library with your entire test. Associating a function library with a test enables you to call any of the functions in the function library from that test.

This chapter includes:

# Introducing Functions

A function is a set of coded steps that perform a particular task for which no suitable method exists by default. You may want your test to include such a task, and even repeat this task several times. Therefore, you want the function to be easily accessible.

For example, you may want to create a step that generates a text file and saves it to the file system, or that accesses data from a Microsoft Excel worksheet.

In this lesson, you will create a function that checks the date format on a page generated by the Mercury Tours Web site, and then add the function to your test. Begin with "Create a Function" below.

# Create a Function

In this exercise, you will create a function that you will later call from your test. This function checks whether a date is displayed in MM/DD/YYYY format. The function also verifies that the date is potentially valid, for example, that the month does not exceed 12 or that the date does not exceed 31.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Select **File > Open > Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Create a new function library.**

   a. Click **New** [ * | ▾ ] down arrow and select **New Function Library**. The New Function Library dialog box opens.

   > **Note:** If you simply clicked the **New** button, and not the **New** button down arrow, the New Test dialog box opens instead.

b. In the New Function Library dialog box, populate the fields as follows:

○ **Look in**: Browse to the **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial** directory, create a new folder named `Tutorial_FunctionLibraries`, and open it to define it as the function library location. (If you copy this path, make sure to replace **<username>** with your username.)

○ **File name**: Enter `CheckDateFunction`.

c. Click **Create**. The blank function library opens in the document pane.

3. **Copy the following function and paste it into the editor.**

```
'The following function checks whether a date string (dateStr)
'has 10 characters representing MM/DD/YYYY

Function check_data_validity( dateStr )
        Dim firstSlashPos, secondSlashPos
        Dim mmPart, ddPart, yyyyPart
    firstSlashPos = inStr( dateStr , "/" )
        secondSlashPos = inStrRev( dateStr, "/" )
        If ( firstSlashPos <> 3 or secondSlashPos <> 6 ) Then
            reporter.ReportEvent micFail,"Format check",
"Date string is missing at least one slash ( / )."
        check_data_validity = False
            Exit function
        End If


        mmPart = mid( dateStr, 1,2 )
        ddPart = mid ( dateStr, firstSlashPos+1, 2 )
         yyyyPart = mid( dateStr, secondSlashPos +1 , 4 )
        If mmPart > 12 Then
             reporter.ReportEvent micFail, "Format Check",
"The month value is invalid. It exceeds 12."
             check_data_validity = False
             Exit function
        End If


        If ddPart > 31 Then
             reporter.ReportEvent micFail, "Format Check",
"The date value is invalid. It exceeds 31."
             check_data_validity = False
             Exit function
         End If


        If yyyyPart < 2000 Then
                reporter.ReportEvent micFail, "Format Check",
"The year value is invalid. (Prior to 2000)"
        check_data_validity = False
             Exit function

         End If

    check_data_validity = True

End Function
```

> **Note:** If you copy and paste the above code as-is, there will be line breaks splitting up the commands at the end of the lines starting with `reporter`. If you try to run this code, the UFT's compiler will generate an error for each instance of this line break. Therefore, at the end of each broken line in the code, press **Delete** to eliminate the break.

4. Click **Save** ![save icon] .

5. **Save the function library.**

6. **Close the function library.**

   Select **File > Close**.

Continue working with your function library, as described in "Associate the Function Library with Your Test" below.

# Associate the Function Library with Your Test

In "Create a Function" on page 45, you created a function library with a single function. In this exercise, you will associate the function library with your test so that its function will be available to your test.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Click the **Open** button down arrow ![open icon], and select **Open Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Associate the CheckDateFunction.qfl with the MercuryTours test.**

   a. If the Solution Explorer is not already open, open it by clicking the **Solution Explorer** button ![solution explorer icon] .

   b. In the Solution Explorer, right-click the **MercuryTours** test node and select **Add > Associate Function Library**. The Open Function Library dialog box opens.

c. Locate and open the CheckDateFunction.qfl function library. In the Automatic Relative Path Conversion dialog box, click **Yes**. This converts the path to a relative path.

The function library is now associated with the **MercuryTours** test, and is displayed in the Solution Explorer as a child of the **MercuryTours** test node.

> **Note:** Using a relative path keeps the path valid when you move folders containing tests and other files from one location to another, as long as the folder hierarchy remains the same.

3. **Save the test.**

Select the **MercuryTours** document tab, and then click **Save** .

All the resources required for your test are now set up, and you can continue to populate your test, as described in

# Chapter 5: Adding Steps

In "Analyzing Your Application and Creating Actions" on page 20, you created unique actions for most of the pages in the Mercury Tours Web site and added them to your test. In other lessons, you created other parts of the required automation infrastructure, such as object repositories and function libraries.

In this lesson, you will add steps to the actions in your test.

This chapter includes:

# Preparing to Add Steps to a Test

Before you add steps to a test, ensure that your application and UFT are set to match the needs of your test. For the purposes of this tutorial:

1. **Set the relevant Internet Explorer options and close all browsers.**

   - In Microsoft Internet Explorer, clear the **AutoComplete** option for all forms, user names, and passwords (select **Tools > Internet Options > Content** tab > AutoComplete **Settings**).

   - Close all browser windows.

2. **Set the relevant UFT Web Page/Frame options.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. If you do not currently have a test open, open a recent test or create a dummy test. To create a dummy test, click **New** [ ✳ ], and create a new test using the default values in the New Test dialog box. You do not need to save this test.

   > **Note:** You must have a test or component open in UFT in order to see the **Web** node in the **Options** dialog box.

   c. Select **Tools > Options > GUI Testing** tab > **Web > Page/Frame Options** node.

   d. In the **Create a new Page test object for** area, select **Different URLs or a change in data transfer** and select the first four check boxes. Click **OK**.

You are now ready to start adding steps to your actions. Begin with "Add Steps to the Login Action" below.

# Add Steps to the Login Action

Each operation performed by UFT on an application is represented by a step in your action. For example, a step might click an image or insert a value into an edit box.

In this exercise, you will make a reservation for a flight from New York to San Francisco on the Mercury Tours Web site, adding steps in various ways. After you add steps to the first action, you will analyze the steps to learn more about the Keyword View.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

b. On the Start page, in the Recent solutions area, click **Tutorial**.

The Tutorial solution opens, displaying the MercuryTours test you created in "Create a New Solution, Test, and Actions" on page 25.

2. **Open the Login action and display the Keyword View.**

   a. If the canvas is not currently open, click the **MercuryTours** tab in the document pane to bring it into focus.

   b. In the canvas, double-click the **Login** action.

   The **Login** action opens.

   c. If the Editor is displayed, click the **Keyword View** button ⬛ to display the Keyword View.

3. **Add the first step to log in to the Mercury Tours Web site.**

   a. In the Keyword View, click in the **Item** column, directly below the **Login** row. The Item list opens, displaying the top-level (parent) test object in the associated object repository, and instructing you to select an item.
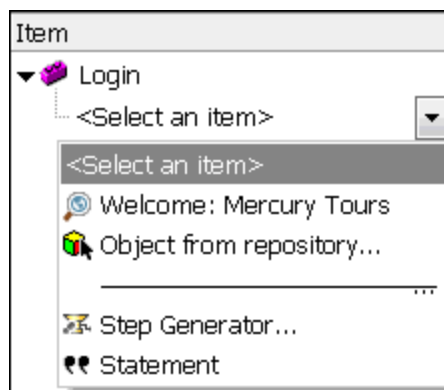
   

   In this step, the parent test object is the **Welcome: Mercury Tours** browser object. Do *not* select this browser object because you do not want to perform an operation on the main browser window. You only need to insert steps on the objects on which you want to perform operations.

   b. Select **Object from repository** to open the Select Test Object dialog box.

c.  Expand the test object tree.
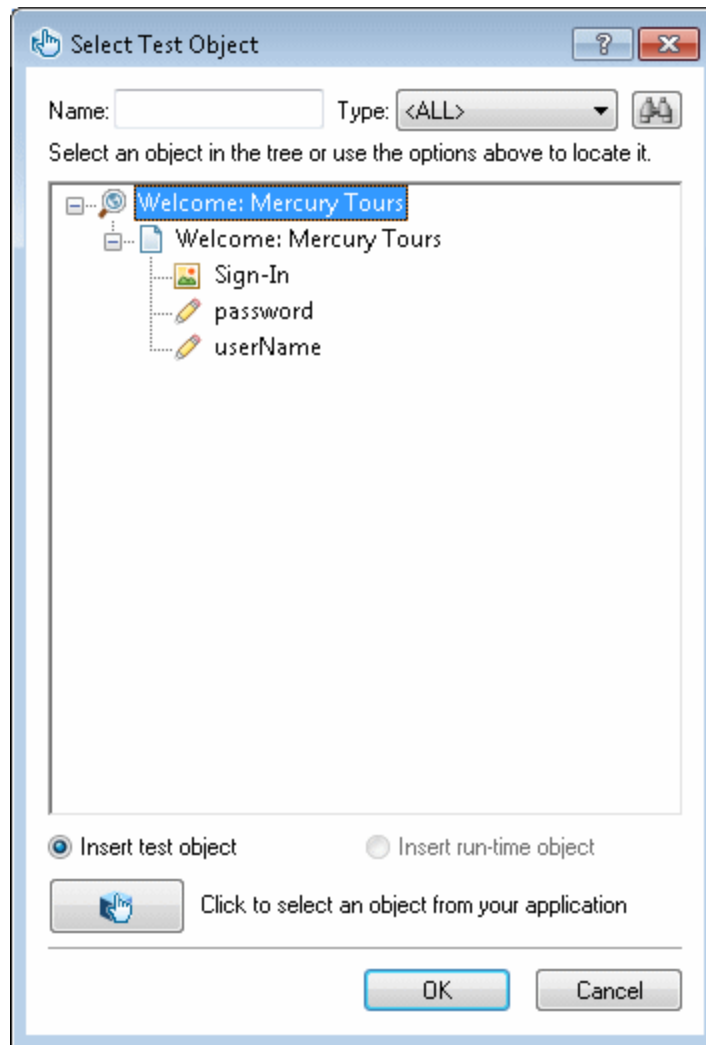


d.  Select **userName** and click **OK**.

The Select Object for Step dialog box closes and one step is added to the action.

Note that three rows are added to the Keyword View. UFT adds a row for each of the parent test objects, even though it does not perform an operation on these objects. These rows are part of the path to the object upon which the step is performed.

During a run session, UFT uses the parent objects to identify the actual object on which it needs to perform an operation.

In this step, represented by the last of the three new rows:

○ The selected **userName** WebEdit test object is added to the **Item** cell.

○ The default method, **Set**, is added to the **Operation** cell.

○ Text is added to the **Documentation** cell indicating that this step clears the text in the edit box. This is because the step is still missing a required value in the **Value** cell, and needs to be updated with the username.

| Item | Operation | Value | Documentation |
|------|-----------|-------|---------------|
| ▼ 🍇 Login | | | |
| ▼ 🔵 Welcome: Mercury Tours | | | |
| ▼ 📄 Welcome: Mercury Tours | | | |
| ✏ userName | Set | | Clear the text in the "userName" edit box. |

e. In the **Value** cell, enter `tutorial`.

Inserting this value completes the step. When you click another area in the Keyword View, the documentation for this step is updated in the **Documentation** cell.

| Item | Operation | Value | Documentation |
|------|-----------|-------|---------------|
| ▼ 🍇 Login | | | |
| ▼ 🔵 Welcome: Mercury Tours | | | |
| ▼ 📄 Welcome: Mercury Tours | | | |
| ✏ userName | Set | "tutorial" | Enter "tutorial" in the "userName" edit box. |

**Tip:** Quotes are automatically added around the value you entered in the **Value** column, indicating that this is a String value. If the method supported an Index value, and you entered an Index value, no quotes would be added.

f. Select **View > Editor** to display the Editor, which displays the syntax of the step in VBScript.

```
Browser("Welcome: Mercury Tours").Page("Welcome: Mercury
Tours").WebEdit("userName").Set "tutorial"
```

Notice that this step is being performed on a **WebEdit** (edit box) test object, named `userName`, and that:

- The test object hierarchy for the WebEdit (edit box) test object is displayed prior to the test object. In this step the hierarchy includes a Browser object and a Page object.

- The method to be performed on the object is displayed immediately after the test object. In this step, the method is **Set**.

- The text to enter in the userName edit box is displayed immediately after the Set method. The text is displayed in quotes, which indicate that this value is a string. In this step, the text to enter is **tutorial**.

- Full stops (periods) separate each part of the step.

g. Click the **Keyword View** button ⊟ to return to the Keyword View.

4. **Add the next step.**

a. Click in the **Item** column, directly below the **userName** row.

The Item list opens, listing the sibling objects of the previous step's test object.

b. Select **password** from the Item list. This time, only one new row is added because the object shares the same parent objects as the previous step.

In this step:

- The **password** WebEdit test object is added to the **Item** cell.

- The default method, **Set**, is added to the **Operation** cell. You need to change this method because the password needs to be encoded.

- Text is added to the **Documentation** cell indicating that this step clears the text in the edit box. This is because the step is still missing a required value in the **Value** cell, and needs to be updated with the password.

c. Click in the **Operation** cell to display the down arrow, and then click the down arrow to display the list of available methods for the selected test object. Select **SetSecure** from the list. This method enables the use of encrypted text. Now you need to generate the encrypted text and insert it in the **Value** cell.

5. **Generate an encoded password using the HP Password Encoder application.**

a. Select **Start > All Programs > HP Software > HP > Unified Functional Testing > Tools > Password Encoder** or `<UFT installation folder>\bin\CryptonApp.exe`. The Password Encoder dialog box opens.
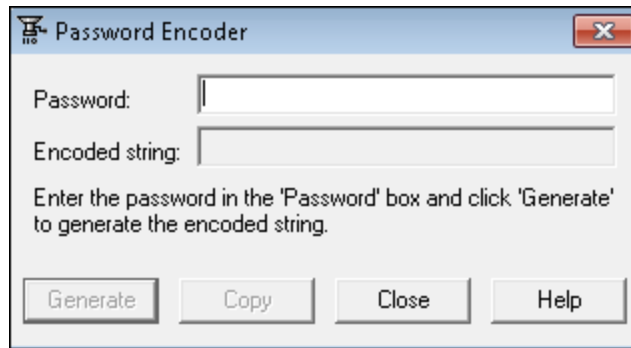
> **Note:** For details on accessing UFT and UFT tools and files in Windows 8, see
> "Accessing UFT in Windows 8 Operating Systems" on page 18.

b. In the **Password** box, enter `tutorial`.

c. Click **Generate**. The Password Encoder encrypts the password and displays it in the
   **Encoded String** box.

d. Click **Copy**. Paste the encoded value in the **Value** cell for the password step.

   Documentation for this step is displayed in the **Documentation** cell.

e. Close the Password Encoder dialog box.

   If you ran your action at this point, UFT would automatically open the Mercury Tours Web
   site and insert the values you specified in the **User Name** and **Password** boxes.

6. **Insert the last step in the Login action.**

   a. Click in the **Item** column below the last step to insert the next step.

      The Item list opens, listing the sibling objects of the previous step's test object.

   b. Select **Sign-In** from the **Item** list.

      This step instructs UFT to click **Sign-In** on the Welcome page and log in to the Mercury
      Tours site.

7. **Save your test.**

   Select **File > Save**.

Learn more about the Keyword View in "Analyzing the Login Action in the Keyword View" on the
next page, and then continue with " Add the Remaining Steps to Your Test" on page 58.

# Analyzing the Login Action in the Keyword View

Let's take a look at the steps in your first action. Each step represents an operation to be performed in the Web browser.

The columns in the Keyword View show different information for each step, as follows:

- **Item.** The item for the step (test object, utility object, function call, or statement) in a hierarchical icon-based tree.

- **Operation.** The operation to be performed on the item, for example, **Click**, **Set**,or **Select**.

- **Value.** The argument values for the selected operation, if required. For example, the text to enter in an edit box, or the mouse button to use when clicking the image.

- **Documentation.** Automatically provided documentation of what the step does, in an easy-to-understand sentence. For example: `Click the "Sign-In" image.`

- **Assignment.** The assignment of a value to or from a variable so you can use the value later in the test. This column is not visible by default.

- **Comment.** Any textual information you want to add regarding the step, for example, `Return to page used in first step of the test`. This column is not visible by default.

> **Note:** You can hide or display individual columns by right-clicking the column heading in the Keyword View, and selecting a column name from the list.
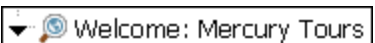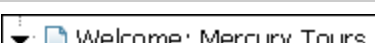
In the **Item** column of the Keyword View, you can click the branch arrows to expand or collapse the steps under each Web page. You can collapse or expand each action by selecting the action name at the top and then selecting **View > Collapse All** or **View > Expand All**.

When you expand your action, it should look similar to this:

| Item | Operation | Value | Documentation |
|------|-----------|-------|---------------|
| ▼ 🧩 Login | | | |
| ▼ 🌐 Welcome: Mercury Tours | | | |
| ▼ 📄 Welcome: Mercury Tours | | | |
| ✏️ userName | Set | "tutorial" | Enter "tutorial" in the "userName" edit box. |
| ✏️ password | SetSecure | "4ffd8cd988f... | Enter the encrypted password in the "password" edit box. |
| 🖼️ Sign-In | Click | | Click the "Sign-In" image. |

Take a few minutes to examine the Keyword View in detail, tracing the steps you performed while creating this action. Then, continue to add additional steps, as described in " Add the Remaining Steps to Your Test" on the next page.

The table below describes the steps shown in the action in the Keyword View above.

| Step | Description |
|---|---|
| ▼ 🐞 Login | **Login** is the action name. Actions are test units that divide your test into logical sections. |
| ▼ 🌐 Welcome: Mercury Tours | The browser object invokes the **Welcome: Mercury Tours** Web site. |
| ▼ 📄 Welcome: Mercury Tours | **Welcome: Mercury Tours** is the name of the Web page. |
| ✏️ userName    Set    "tutorial" | **userName** is the name of the edit box. **Set** is the method performed on the edit box. **tutorial** is the value entered in the userName edit box. |
| ✏️ password    SetSecure    "4f8eba5... | **password** is the name of the edit box. **SetSecure** is an encryption method performed on the edit box. The list of digitsis the encrypted value entered in the password edit box. |
| 🖼️ Sign-In    Click | **Sign-In** is the name of the image link. **Click** is the method performed on the image. |

For more details on the Keyword View, see the Keyword View chapter in the *HP Unified Functional Testing User Guide*.

# Add the Remaining Steps to Your Test

Now that you added steps using the Keyword View in "Add Steps to the Login Action" on page 51, you can continue with other methods of adding steps, as follows:

- "Add Steps to the FlightFinder Action by Recording" below

- "Add a Step to the SelectFlight Action Using the Toolbox Pane" on page 63

- "Add Steps to the BookFlight Action Using the Editor and Step Generator" on page 64

# Add Steps to the FlightFinder Action by Recording

In "Add Steps to the Login Action" on page 51, you added steps to perform on the Mercury Tours Welcome page. If you were to run your test at this point, the last step would open the Flight Finder page in the Mercury Tours Web site.

In this exercise, you will record steps for the **FlightFinder** action you created for the Flight Finder page. The steps you add will use test objects from the associated **MercuryToursFlightFinder** shared object repository.

> **Tip:** Before you begin a recording session, you may want to place the browser window and this tutorial window side-by-side on your screen. This allows you to read the tutorial during recording.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Select **File > Open > Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Open the FlightFinder action.**

   a. If the canvas is not currently open, click the **MercuryTours** tab in the document pane to bring it into focus.

   b. In the canvas, double-click the **FlightFinder** action. The FlightFinder action opens.

3. **Launch the Mercury Tours Web site and open the Flight Finder page.**

   a. In Microsoft Internet Explorer, browse to the Mercury Tours Web site: http://newtours.demoaut.com

   The Mercury Tours home page opens.

b. In the Find a Flight area, enter `tutorial` in both the **User Name** and **Password** boxes and click **Sign-In**. The Flight Finder page opens.
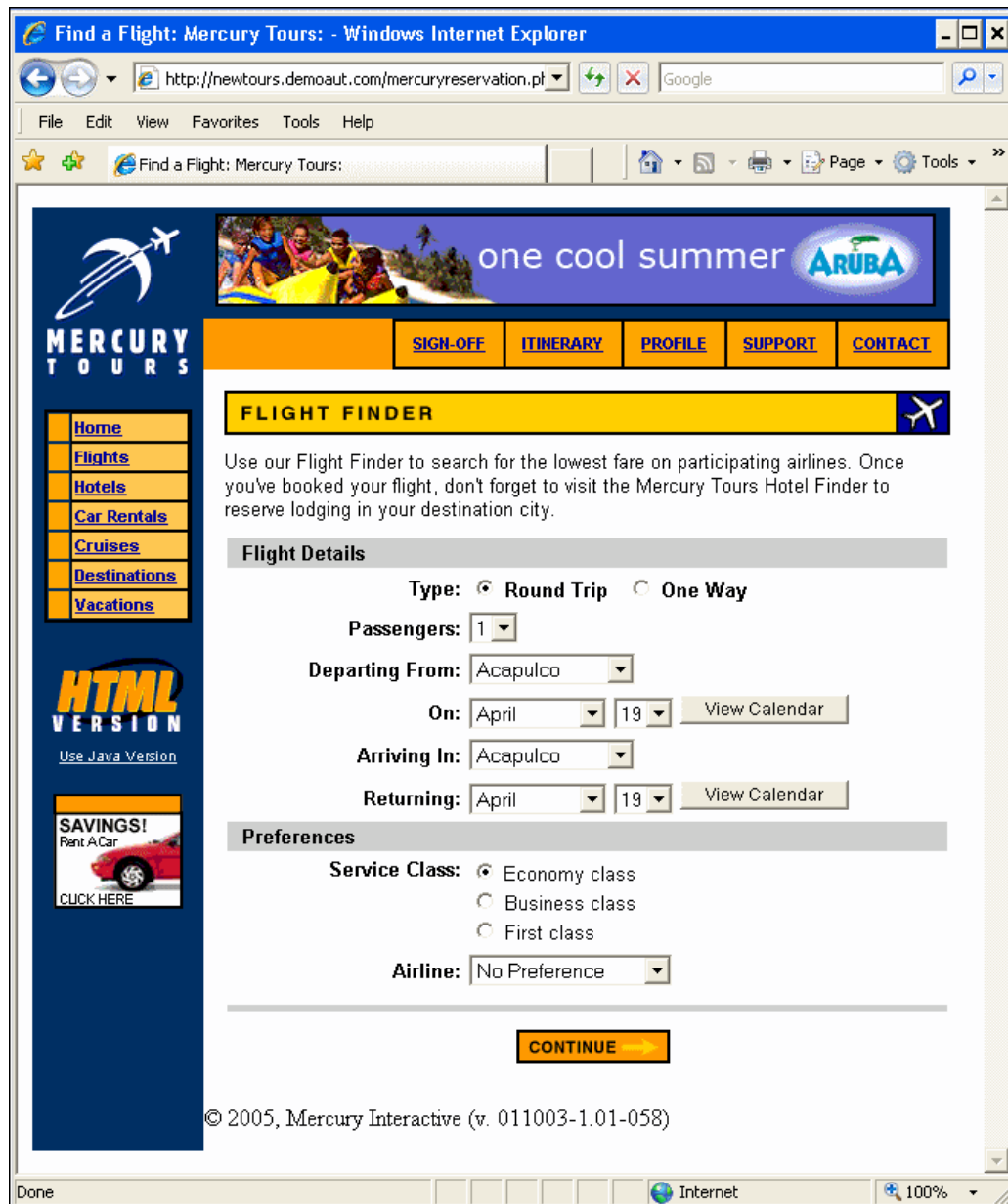


c. In UFT, select **Record > Record and Run Settings**. The Record and Run Settings dialog box opens.

In the **Web** tab, select **Record and run test on any open browser** and click **OK**. This enables UFT to record on the open Web page.
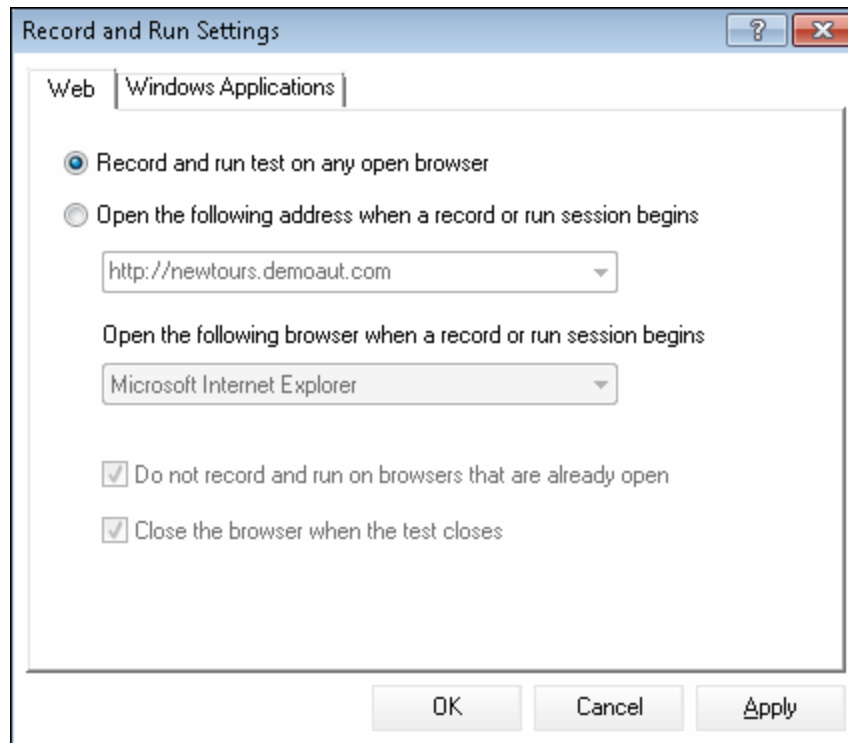
d.  In UFT, click the **Record** button ⏺ . The recording toolbar appears, and UFT begins recording.

    The UFT window disappears, and your browser displays the Flight Finder page in the Mercury Tours Web site.

e.  In the Mercury Tours Flight Finder Web page, change the following selections by selecting from the drop-down lists. (Do not use the calendar option.)

    ○  Departing From: **New York**

    ○  On: **December 29**

    ○  Arriving In: **San Francisco**

    > **Note:** If you are performing this tutorial in the month of December, or on the 29th or the 31st of any month, select a different month or date while recording. UFT records an operation only when you make a change in the application, so this ensures that the step is recorded. If you accept a default value (or re-select the default value), UFT does not record an operation.

f. Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

g. On the **Recording** toolbar, click **Stop** 🔲 to stop the recording process.

You have now reserved an imaginary ticket from New York to San Francisco. UFT recorded your Web browser operations from the time you clicked the **Record** button in UFT until you clicked **Stop** on the Recording toolbar.

h. In UFT, select **Record > Record and Run Settings** to restore the required run settings**.** The Record and Run Settings dialog box opens.

In the **Web** tab:

○ Select **Open the following address when a record or run session begins**.

○ Confirm that the URL in the first box is http://newtours.demoaut.com.

○ In the second box, select a browser on which to run your test session. Microsoft Internet Explorer is the browser used in this tutorial.

○ Confirm that **Do not record and run on browsers that are already open** and **Close the browser when the test closes** are selected.

○ Click **OK** to save your changes and close the dialog box.
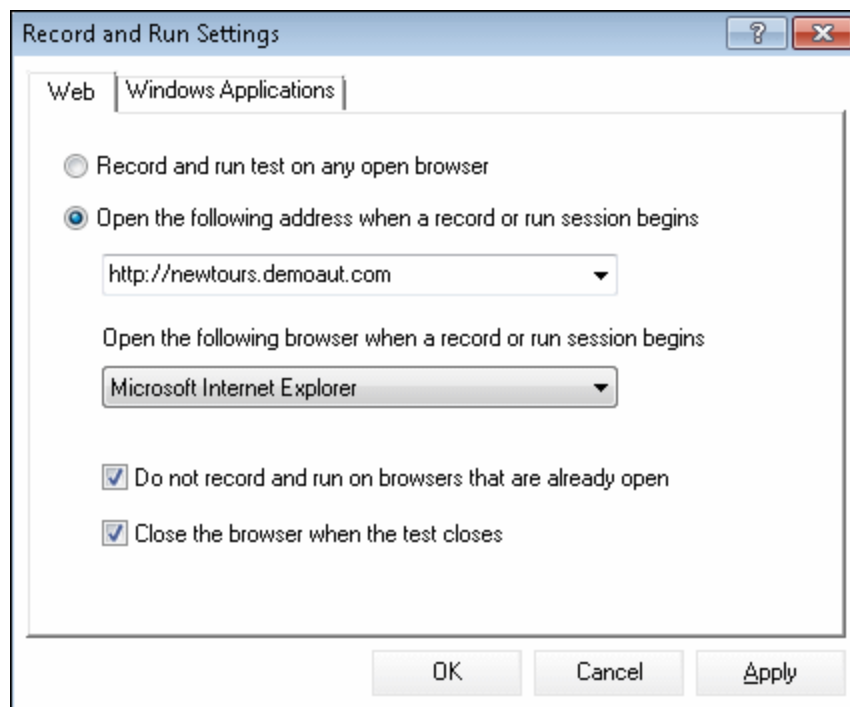
4. **Save your test.**

Click **Save** 💾 .

Do not close the test, because you still want to add steps to other actions. Continue with "Add a Step to the SelectFlight Action Using the Toolbox Pane" below.

# Add a Step to the SelectFlight Action Using the Toolbox Pane

In "Add Steps to the FlightFinder Action by Recording" on page 58, you added steps to perform on the Mercury Tours Flight Finder page. If you were to run your test at this point, the last step would open the Select Flight page in the Mercury Tours Web site.

For the purpose of this tutorial, you do not need to make any changes in the Select Flight page, but you do need to instruct UFT to accept the default options and continue to the next page. Therefore, in this exercise, you will add one step that instructs UFT to click **CONTINUE**. You will drag the relevant test object into your action from the Toolbox Pane, which is another way of creating steps in your action.

> **Note:** Toolbox items are listed according to the action in focus in the document pane. If a test or function library is in focus, or if you do not have a test open at all, the Toolbox Pane is empty.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Click the **Open** button down arrow 📂 , and select **Open Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Open the SelectFlight action.**

   In the Solution Explorer, double-click the **SelectFlight** action. If the Solution Explorer is not already open, open it by clicking the **Solution Explorer** button 🗔 .

   The **SelectFlight** action opens.

3. **Display the Toolbox Pane.**

   Click the **Toolbox** button ![icon]. .

   The Toolbox pane works in both the Editor and the Keyword View, so there is no need to switch.

4. **Select the object that represents the CONTINUE button on the Select Flight Web page.**

   a. In the Toolbox Pane, expand the Test Objects tree.

   b. Locate and drag the **reserveFlights** Image ![icon] object into the action, in either the Editor or the Keyword View. (The **reserveFlights** Image object should be close to the top of the list of objects in the Toolbox Pane.)

   > **Note:** You may notice that this object repository has many more objects than the **MercuryToursLogin** repository. This is because you did not delete extraneous objects in "Creating Object Repositories" on page 30.

   The selected Image test object is added to the step, together with its default method, **Click**.

   ○ In the Keyword View, the **Documentation** cell describes what the step does in plain English. Again, this step is displayed on three rows in the Keyword View because the parent test objects are part of the step.

   ○ In the Editor, the step is displayed as follows:

   ```
   Browser("Select a Flight: Mercury").Page("Select a Flight:
   Mercury").Image("reserveFlights").Click
   ```

5. **Save your test.**

   Select **File > Save**.

   Do not close the test, because you still want to add steps to other actions. Continue to add steps, as described in "Add Steps to the BookFlight Action Using the Editor and Step Generator" below.

# Add Steps to the BookFlight Action Using the Editor and Step Generator

In "Add a Step to the SelectFlight Action Using the Toolbox Pane" on the previous page, you added a step to perform on the Mercury Tours Select Flight page. If you were to run your test at this point, the last step would open the Book a Flight page in the Mercury Tours Web site.

In this exercise, you will create steps by copying and pasting content into the Editor, which is another way of creating steps in your action. The pasted steps do not include values. You will then insert values for each step using the Keyword View.

Finally, you will use the Step Generator to insert a single step. The Step Generator enables you to define an entire step in one dialog box, instead of inserting different parts of a step in the various columns of the Keyword View.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. On the Start page, in the Recent solutions area, click **Tutorial**.

      The Tutorial solution opens, displaying the MercuryTours test you created in "Create a New Solution, Test, and Actions" on page 25.

2. **Open the BookFlight action.**

   In the Solution Explorer, double-click the **BookFlight** action. (If the Solution Explorer is not already open, open it by clicking the **Solution Explorer** button 🖼️ .)

   The **BookFlight** action opens.

3. **Add steps to the BookFlight action.**

   a. Click the **Editor** button [icon] to display the Editor.

   b. Copy the following steps and paste them into the editor:

```
Browser("Book a Flight: Mercury").Page("Book a Flight:
Mercury").WebEdit("passFirst0").Set
Browser("Book a Flight: Mercury").Page("Book a Flight:
Mercury").WebEdit("passLast0").Set
Browser("Book a Flight: Mercury").Page("Book a Flight:
Mercury").WebList("creditCard").Select
Browser("Book a Flight: Mercury").Page("Book a Flight:
Mercury").WebEdit("creditnumber").Set
Browser("Book a Flight: Mercury").Page("Book a Flight:
Mercury").WebList("cc_exp_dt_mn").Select
Browser("Book a Flight: Mercury").Page("Book a Flight:
Mercury").WebList("cc_exp_dt_yr").Select
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").Image
("buyFlights").Click
```

   **Note:** If you copy and paste the above code as-is, there will be line breaks splitting up the commands at the end of each line. If you try to run this code, the UFT's compiler will generate an error for each instance of this line break. Therefore, at the end of each line in the code, press **Delete** to eliminate the break.

4. **Fill in the missing step details.**

a.  Select **View > Keyword View** to display the Keyword View.

b.  Look at the **Documentation** column. Notice that for each step, either no documentation is available, or the documentation indicates that the step is clearing a text box. This is because the steps you just inserted are missing values. For example, in the first step, a value needs to be filled in for the passenger's first name. The details that you need to add are encircled in the following example.



c.  In the **Value** cell for the relevant step in the Keyword View, enter the following:

| For: | Enter: |
| --- | --- |
| passFirst0 | Your first name (or a fictitious first name) |
| passLast0 | Your last name (or a fictitious last name) |

| For: | Enter: |
|------|--------|
| creditCard | Any of the following credit card names:<br><br>○ American Express<br><br>○ MasterCard (one word)<br><br>○ Visa<br><br>○ Discover<br><br>○ Diners Club<br><br>○ Carte Blanche |
| creditnumber | Any 8-digit number. (This is a fictitious credit card number.) |
| cc_exp_dt_mn | Any month from **01** to **12**. Make sure that you enter the value as a two-digit number. (This is the month part of the credit card expiration date.) |
| cc_exp_dt_yr | Any year from **2008** to **2010**, as a four-digit number. (This is the year part of the credit card expiration date.) |

**Note:** Notice that quotation marks are added to the values you enter in the **Value** column after you click outside of the cell. These quotes are necessary for the test to run correctly, and indicate to UFT that the value is a string. If you go back and change a value, you must add these quotation marks manually, because they are not automatically added a second time.
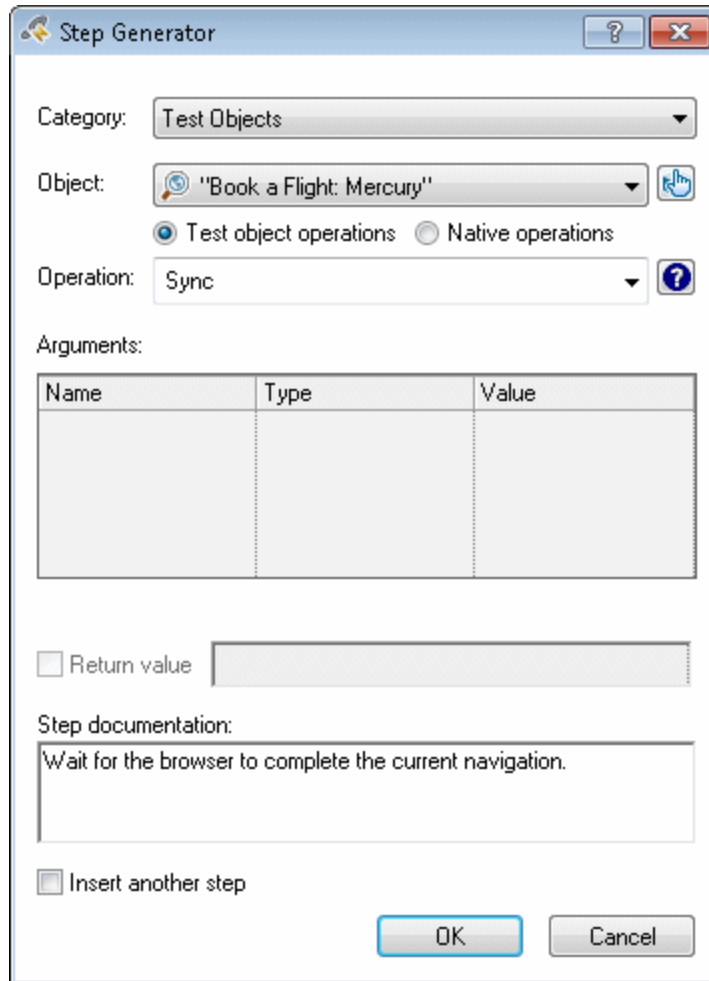
5. **Add a step using the Step Generator to confirm the booking and return to the Home page.**

The last step that you pasted into the BookFlight action clicks **SECURE PURCHASE** in the Book a Flight page to complete the transaction. During a run session, this opens the Flight Confirmation page in the Mercury Tours Web site.

You will now use the Step Generator to define a step to return to the Mercury Tours Welcome page.

a. Click in the **Item** column below the last step to ensure that the next step will be inserted after the last existing step in this action.

b. Scroll down to the bottom of the **Item** list and select **Step Generator**. The Step Generator dialog box opens.
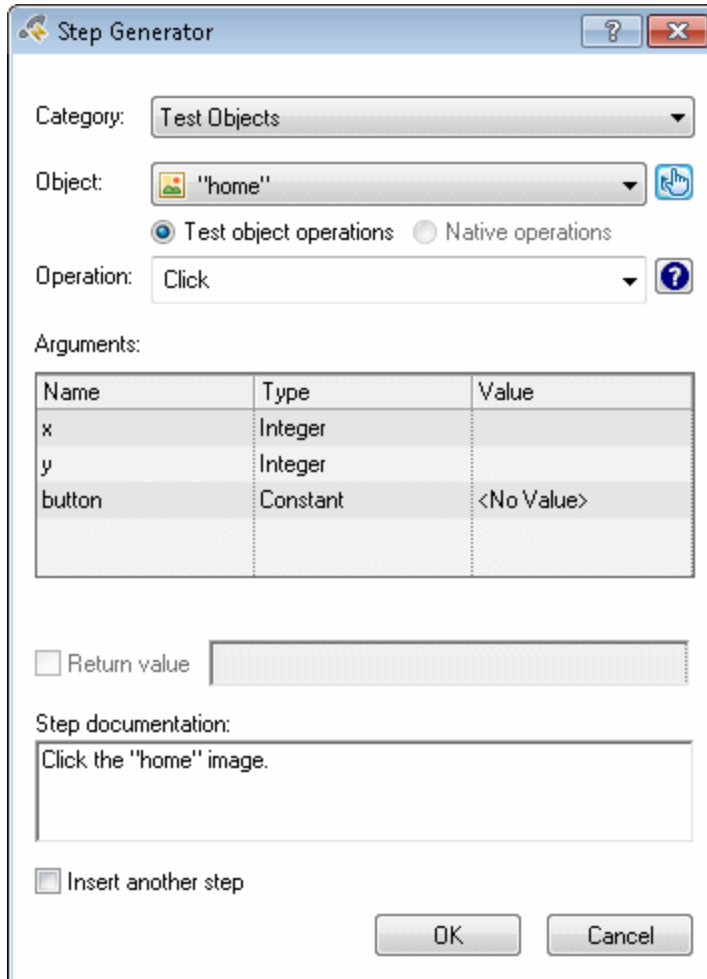


c. In the **Object** drop-down, select **"Flight Confirmation: Mercury"**.

d. Click the **Select Object** button ⬚ . The Select Test Object dialog box opens. Expand the **Flight Confirmation: Mercury** node of the tree.

> **Note:** You may want to collapse the **Book a Flight: Mercury** node of the tree to view the **Flight Confirmation: Mercury** node.
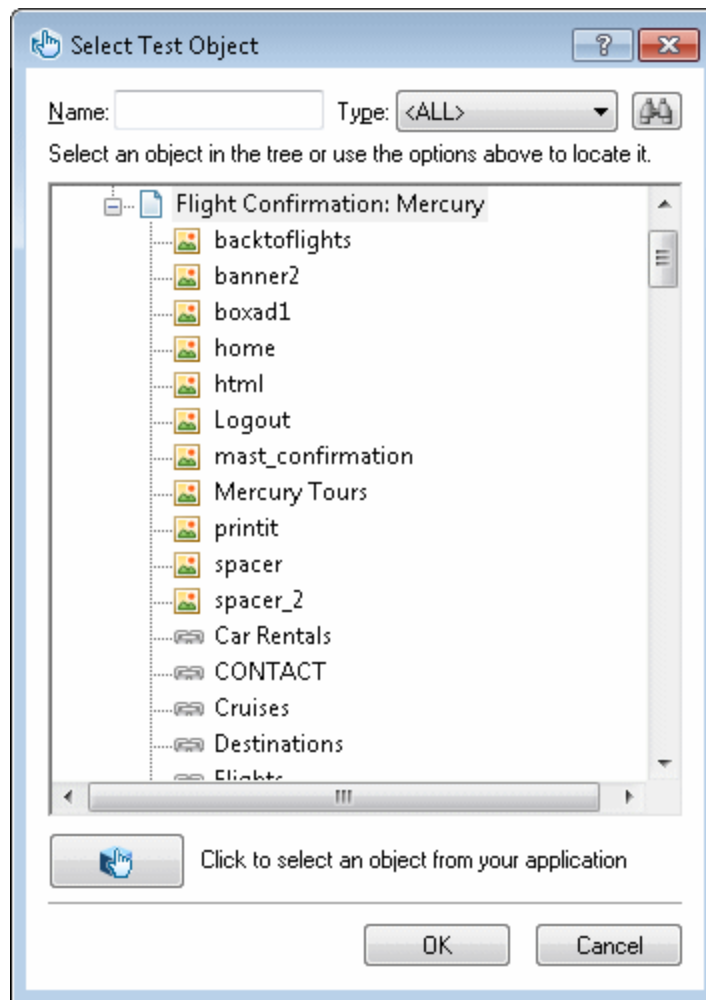
e. Select the **home** Image object ![icon], and click **OK**. The Step Generator displays the default options for the **home** test object.



Define the arguments and values in the Step Generator just as you would in the Keyword View columns:

○ Use the **Category** and **Object** drop-down lists to define the value in the **Item** column in the Keyword View.

○ Use the **Operation** drop-down lists to define the value in the **Operation** column. In this step, the default operation for the **home** test object, **Click**, is displayed in the **Operation** box. Although you could potentially select a different operation for this test object by clicking the down arrow, **Click** is the operation needed for this step.

○ Use the **Arguments** table to define values for the arguments in the **Value** cell in the Keyword View. If any of the arguments were mandatory, a red asterisk would be displayed next to the argument name. In this step, there are no mandatory arguments,

and you do not need to define any values.

- ○ The **Step documentation** box displays the instruction for this step as it will be displayed in the **Documentation** cell of the Keyword View.

- ○ If you want to insert another step after this one, select the **Insert another step** check box to open the Step Generator dialog box again after adding this step. In this action, this is the only step you need in, so there is no need to select this check box.

f.  Click **OK**. The Step Generator closes and the step is added to the Keyword View.

g.  Click the **Editor** button [icon] to display the Editor.

h.  You can see that the same step is displayed as follows:

```
Browser("Flight Confirmation: Mercury").Page("Flight Confirmation:
Mercury").Image("home").Click
```

6.  **Save your test.**

Click **Save** [icon] .

Congratulations! You have created your first test and are ready to run it. Continue with .

# Chapter 6: Running and Analyzing Tests

In "Adding Steps " on page 50, you added steps to each of your test actions, and you are now ready to run your test.

When you run your test in this lesson, UFT opens the appropriate application and performs each step in the test sequentially. When UFT finishes running the test, it displays the results of the run.

This chapter includes:

# Run a Test

In this exercise, you will run the test you prepared in "Adding Steps " on page 50.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Select **File > Open > Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Configure UFT to save all images to the run results.**

   UFT allows you to determine when to save images to the run results. In this lesson, all images should be saved to the run results.

   a. Select **Tools > Options > GUI Testing** tab > **Screen Capture** node.

   b. Select the **Save still image captures to results** checkbox, and then select **Always** from the drop-down menu.

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

3. **Start running your test.**

   a. In the Solution Explorer, select the MercuryTours test.

   b. Click the **Run** button . The Run Test dialog box opens.

c. Click the Options bar to expand the Run Test Options area, and verify that **New run results folder** is selected. Accept the default results folder name.
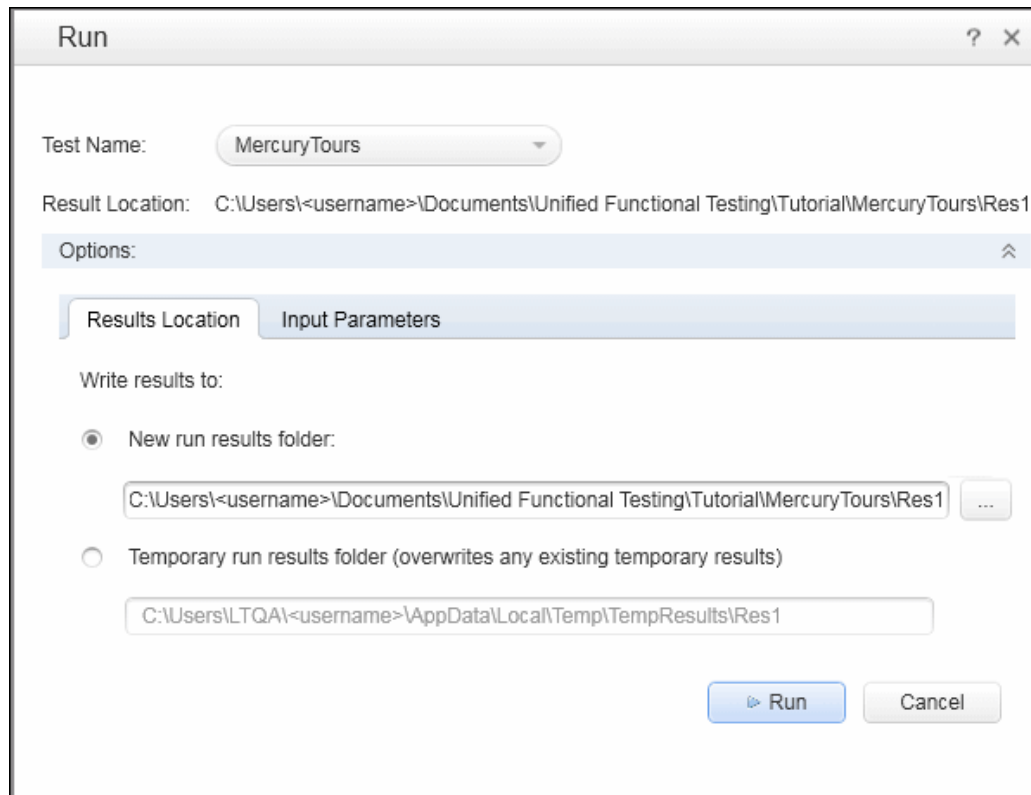


d. Click **Run** to close the Run dialog box and start running the test.

Watch carefully as UFT opens your browser and starts running the test. In the browser, you can see UFT perform each step you inserted; a yellow arrow in the left margin of the Keyword View and the highlighted row indicate the step that UFT is running.
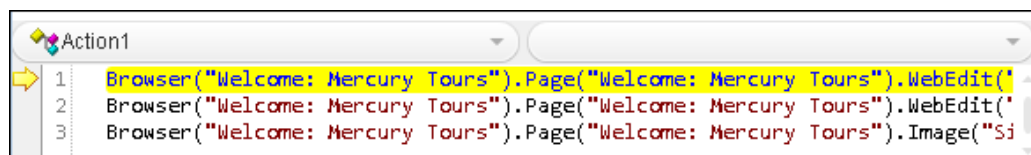


If any errors appear, go to the point in the test that is indicated in the error message, and verify that the step is configured as described in the relevant task in this tutorial.

When the test run is complete, learn more about the run results, as described in "Navigating the Run Results Viewer" on the next page and "Analyze the Run Results" on page 77.

# Navigating the Run Results Viewer

When UFT finishes running the test, the Run Results Viewer opens. Drag the pane borders as needed to clearly view all populated panes.
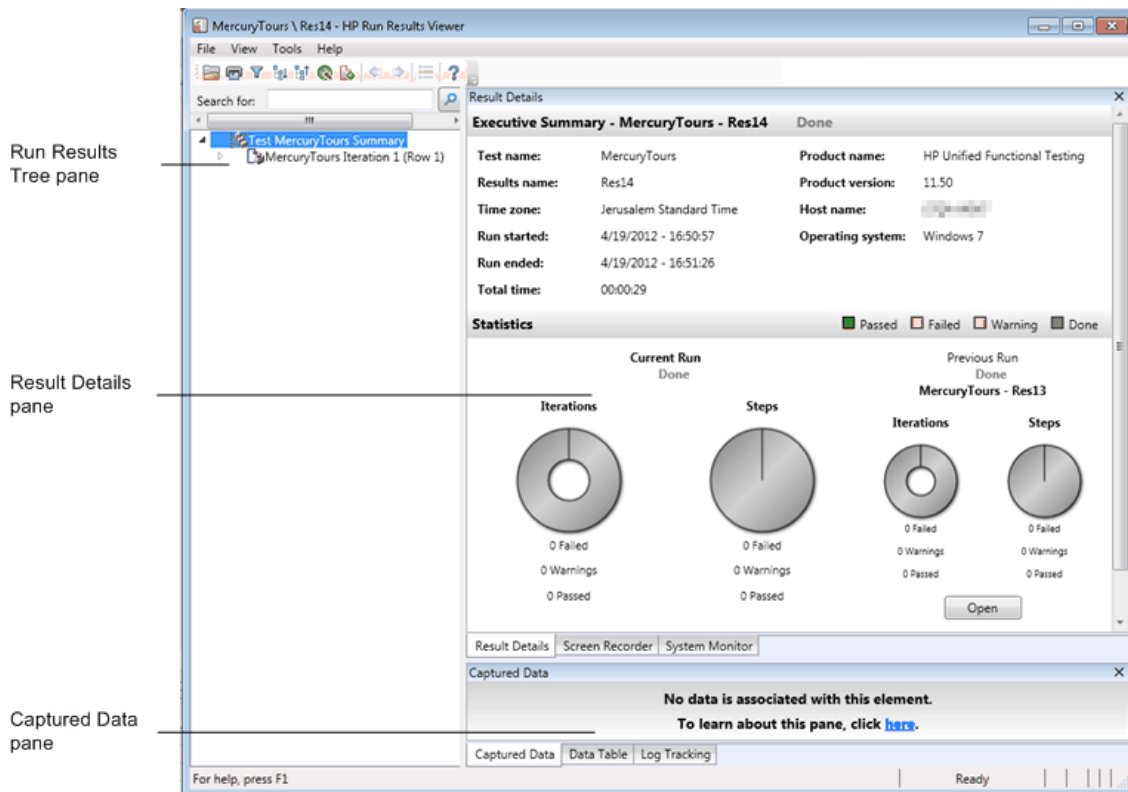


Initially, the Run Results Viewer displays the following panes:

- **Run Results Tree pane.** A graphical representation of the results in an expandable tree, organized according to the actions and the Web pages visited during the test run. The steps performed during the run are represented by icons in the tree, which can be expanded (**arrow**) to view each step. You can instruct UFT to run a test or action more than once using different sets of data in each run. Each run is called an **iteration**, and each iteration is numbered. (The test you ran had only one iteration.)

- **Result Details pane.** A high-level results overview report, containing general information about the test, which steps passed or failed, statistics in pie chart format, and so on.

- **Captured Data pane.** A still image of the state of your application at a particular step. This pane is only populated when you select a step from the Run Results tree, and the selected step has a recorded image.

> **Note:** The Test Flow pane is an additional pane in the Run Results Viewer, and is relevant

for API tests only. This pane is not populated for the current GUI test, and you can close it. For details about API testing, see the *HP Unified Functional Testing User Guide*.

Your test run succeeded because UFT was able to navigate the Mercury Tours site according to the steps that you added. If an error occurred and your test did not run successfully, the error will be shown in the Run Results Viewer. In such cases, go back and make sure that the steps are configured exactly as described in this tutorial.

# Analyze the Run Results

In this exercise, you will inspect the steps UFT performed when running your test in "Run a Test" on page 74. You can view snapshots of the application window for each step.

1. **View the results for a specific step.**

   In the results tree, use the arrow to expand the **MercuryTours Iteration 1 (Row 1) > FlightFinder Summary > Find a Flight: Mercury > Find a Flight: Mercury** node to display all of the steps performed on the Flight Finder page.

   In the run results tree, select **fromPort:Select**.
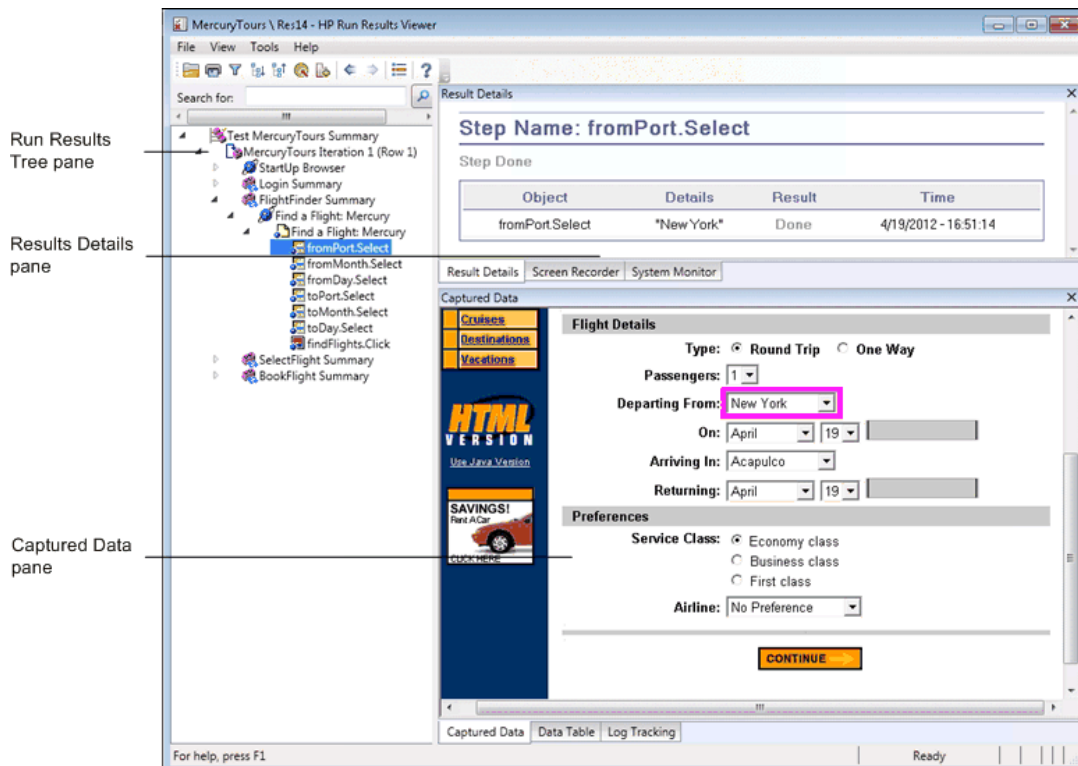
The Run Results Viewer now displays the following information:

- The Run Results Tree pane, with one step highlighted.

- The Result Details pane, displaying details of the highlighted step.

- The Captured Data pane, showing a screen capture of the Web page on which the step was performed UFT.

When you click a page in the results tree, UFT displays the corresponding page in the Captured Data pane. When you click a step (an operation performed on an object) in the results tree, the corresponding object is highlighted in the Captured Data pane. In this case, the **Departing From** text box is highlighted.

2. **Close the Run Results Viewer.**

   Select **File** > **Exit**.

You have now successfully created and run a test that reserves a flight from New York to San Francisco on the Mercury Tours Web site. You can enhance your test by continuing with the following lessons:

- "Creating Checkpoints and Using Functions" on page 79

- "Parameterizing Steps and Objects" on page 114

# Chapter 7: Creating Checkpoints and Using Functions

In "Running and Analyzing Tests" on page 73, you ran a test that you'd created in previous lessons, to check that a series of steps performed on the Mercury Tours Web site runs smoothly.

Your test is now ready for additional enhancements, including checkpoints. A checkpoint verifies that expected information is displayed in your application while the test is running. In this lesson, you will insert checkpoints and use a function to check the validity of the some of the objects in the Mercury Tours Web site.

This chapter includes:

# Understanding Checkpoint Types

UFT offers the following types of checkpoints:

| Checkpoint Type | Description | Example of Use |
|---|---|---|
| **Standard Checkpoint** | Checks values of an object's properties. | Check that a radio button is selected. |
| **Image Checkpoint** | Checks the property values of an image.<br><br>(Select the Standard Checkpoint option and then selecting to check a Web Image object.) | Check that the image source file is correct. |
| **Table Checkpoint** | Checks information in a table.<br><br>(Select the Standard Checkpoint option and then selecting to check any table object.) | Check that the value in a table cell is correct. |
| **Page Checkpoint** | Checks the characteristics of a Web page.<br><br>(Select the Standard Checkpoint option and then selecting to check a Web Page object.) | Check how long a Web page takes to load or if a Web page contains broken links. |
| **Text Checkpoint** | Checks that a text string is displayed in the appropriate place in an application. | Check whether the expected text string is displayed in the expected location in a test object. |
| **Text Area Checkpoint** | Checks that a text string is displayed within a defined area in a Windows-based application. | Check that an area of a dialog box includes text that was entered in another part of the application. |
| **Bitmap Checkpoint** | Checks an area of an application after capturing it as a bitmap. | Check that a Web page (or any portion of it) is displayed as expected. |
| **Database Checkpoint** | Checks the contents of databases accessed by an application or Web site. | Check that the value in a database query is correct. |

| Checkpoint Type | Description | Example of Use |
|---|---|---|
| **Accessibility Checkpoint** | Identifies areas of a Web site to check for Section 508 compliance. | Check if the images on a Web page include ALT properties, required by the W3C Web Content Accessibility Guidelines. |
| **File Content Checkpoint** | Checks the text in a document generated or accessed during a run session. | Check that the headers in a dynamically-generated PDF file display the regional corporate headquarters contact information correctly. |
| **XML Checkpoint** | Checks the data content of XML documents. | Check the content of an element to make sure that its tags, attributes, and values have not changed.<br><br>**Note:** XML file checkpoints are used to check a specified XML file; XML application checkpoints are used to check an XML document within a Web page. |

You can add most checkpoints to your test either while inserting steps or while recording. The following sections explain how to create some of the checkpoints described above, in the test you created in "Add Steps to the Login Action" on page 51.

**Note:** When UFT creates a checkpoint, it assigns a name based on information inside the checkpoint—the checked value, for example. The checkpoint name remains unchanged, even if you subsequently modify the information on which it was based. Keep this in mind when looking for checkpoints displayed in the Keyword View. Note also, that UFT may shorten the name displayed in the Keyword View.

For additional details about checkpoints, see the *HP Unified Functional Testing User Guide*. To add checkpoints to your test, begin with "Create a Checkpoint Test" below.

# Create a Checkpoint Test

In this exercise, you will save the MercuryTours test that you ran in "Running and Analyzing Tests" on page 73 as a new test to create your checkpoints.

Checkpoints do not need to be managed in separate tests, and you are only creating a new test now for the sake of the tutorial. During your regular working process, you can add checkpoints to any test.

1. **Start UFT and open the MercuryTours test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Click the **Open** button down arrow

, and select **Open Solution**. The Open Solution dialog box opens.

    c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Save the test as Checkpoint.**

    a. In the Solution Explorer, select the **MercuryTours** test node, and then select **File > Save As**.

    b. In the Save Test As dialog box, browse to the **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial** directory, and save the test as `Checkpoint`.

    In the Solution Explorer, the MercuryTours test is replaced by the new Checkpoint test. The MercuryTours test is still saved separately in the file system.

3. **Add the MercuryTours test back to the solution.**

You can have both the MercuryTours and the Checkpoint tests open at the same time if they are both referenced from the same solution. This enables you switch back and forth between them if you want to compare or edit the tests. You can only run a single test at a time.

    a. Select **File > Add > Existing Test**.

    b. Navigate to the **MercuryTours** test located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Add**. (You created the MercuryTours test in "Create a New Solution, Test, and Actions" on page 25.)

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

    c. The MercuryTours test appears in the Solution Explorer. Tests are organized in the Solution Explorer alphabetically.

The solution is saved automatically. Continue with "Check Object Values" on the next page.

# Check Object Values

In this exercise, you will add a standard checkpoint to the test you created in "Create a Checkpoint Test" on page 81, in the Book a Flight action. This checkpoint verifies the value entered in the box containing the passenger's first name.

> **Note:** The Mercury Tours Web site must be open to the page you want to check before you insert the checkpoint.

1. **Start UFT and open the Checkpoint test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. On the Start page, in the Recent Solutions area, click **Tutorial**. The Tutorial solution opens.

   c. In the Solution Explorer, select the **Checkpoint** test node. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

2. **Display the action in which you want to add a checkpoint.**

   You want to add a checkpoint that checks the property values of the **First Name** edit box after the test automatically enters the passenger's name in the box.

   In the canvas, double-click the **BookFlight** action to open it.

3. **Open the Mercury Tours application to the Book a Flight page.**

   a. Log in to the Mercury Tours Web site: http://newtours.demoaut.com

   In the **User Name** and **Password** boxes, enter `tutorial`.

   Click **Sign-In**. The Flight Finder page opens.

   b. Enter flight details.

   Change the following selections by selecting from the drop-down lists. (Do not use the calendar option.)

   - Departing From: **New York**

   - On: **December 29**

   - Arriving In: **San Francisco**

   - Returning: **December 31**

> **Note:** If you are performing this tutorial in the month of December, or on the 29th or the 31st of any month, select a different month or date while recording. UFT records an operation only when you make a change in the application, so this ensures that the step is recorded. If you accept a default value (or re-select the default value), UFT does not record an operation.

    c. Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

    d. In the Select Flight page, click **CONTINUE** to accept the default flight selections. The Book a Flight page opens.

4. **Create a standard checkpoint.**

    a. If the Editor is displayed, click the **Keyword View** button to display the Keyword View.

    b. In the **Item** column, select the **passFirst0** row.

c.  Select **Design > Checkpoint > Standard Checkpoint**. The Checkpoint Properties dialog box opens.
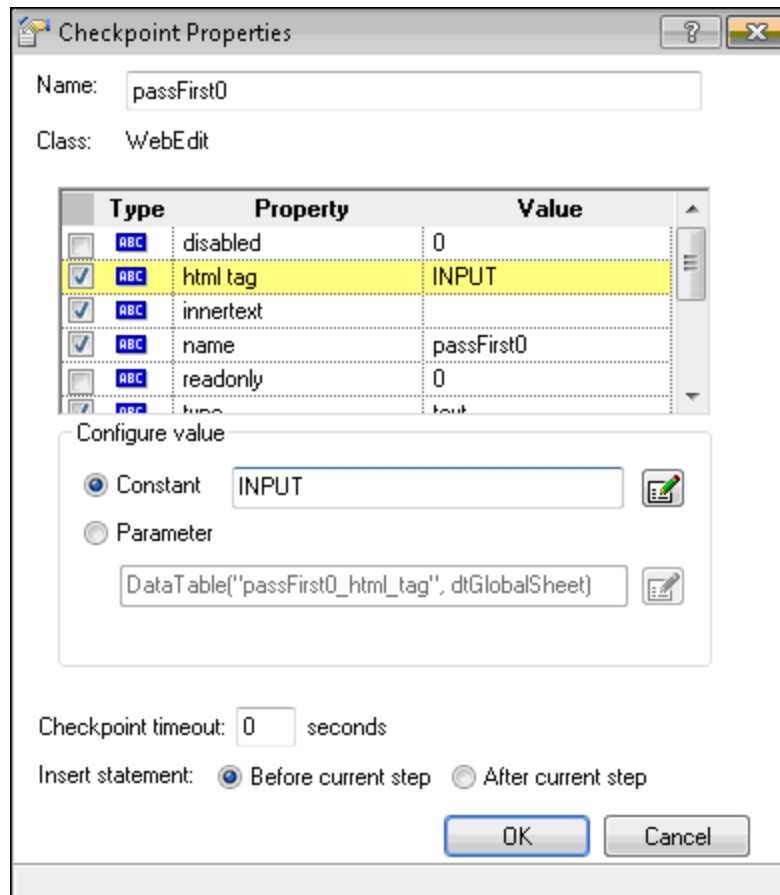


The dialog box displays the object's properties:

○   The **Name** is the name of the object as defined in the HTML code of the Web page, in this case **passFirst0**.

○   The **Class** is the type of object, in this case **WebEdit,** indicating that the object is an edit box.

○   The **ABC** icon in the **Type** column indicates that the value of the property is a constant.

UFT recommends default property checks for each object class,. The table below describes the default checks recommended for the WebEdit class.

| Property | Value | Explanation |
|----------|-------|-------------|
| html tag | INPUT | **INPUT** is the html tag as defined in the HTML source code. |

| Property | Value | Explanation |
|----------|-------|-------------|
| innertext | | In this case, the value of innertext is empty. The checkpoint checks that the value is empty. |
| name | passFirst0 | **passFirst0** is the name of the edit box. |
| type | text | **text** is the type of object as defined in the HTML source code. |
| value | | Currently, the value is empty. You need to enter the same value that you specified for the **First Name** edit box. |

d. In the **Name** box of the Checkpoint Properties dialog box, enter CheckName as the new checkpoint name.

e. Scroll down in the object properties area and select the row containing the property named **value**. You need to enter the value that matches the value you entered in the actual step. To do this, click in the **Constant** box and enter the value you defined as the passenger's first name, above in "Add Steps to the BookFlight Action Using the Editor and Step Generator" on page 64.

> **Note:** You can see what you entered by moving the Checkpoint Properties dialog box so that you can view the Keyword View. The value that you entered is listed in the Value column of the passFirst0 row, in quotes.

f. In the **Insert statement** area at the bottom of the Checkpoint Properties dialog box, select **After current step**. This inserts the checkpoint after the **passFirst0 Set**... step. (This is the step in which the passenger's first name is entered in the **First Name** box.)

g. Accept the rest of the settings as default and click **OK**. UFT adds a standard checkpoint step to your test below the selected step.

| | | | |
|---|---|---|---|
| ▼ 📄 Book a Flight: Mercury | | | |
| ✏️ passFirst0 | Set | "Nicole" | Enter "Nicole" in the "passFirst0" edit box. |
| ✏️ passFirst0 | Check | CheckPoint("CheckName") | Check whether the "passFirst0" edit box h |
| ✏️ passLast0 | Set | "Jones" | Enter "Jones" in the "passLast0" edit box. |

5. **Save the test.**

   Click **Save** 💾 .

You can use this process to insert many checkpoints. Continue to enhance your test by adding a page checkpoint, as described in "Check Pages" on the next page.

# Check Pages

In this exercise, you will add a page checkpoint to the test you created in "Create a Checkpoint Test" on page 81. The page checkpoint checks that the number of links and images in the page when you run your test is the same as when UFT learned the objects on which you inserted steps in your test.

1. **Start UFT and open the Checkpoint test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Select **File > Open > Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

   d. In the Solution Explorer, select the **Checkpoint** test node.

2. **Locate the step where you want to add a page checkpoint.**

   a. If the Checkpoint BookFlight action is not already open, in the Solution Explorer, double-click the **BookFlight** action. The BookFlight action opens.

   b. In the **Item** column of the Keyword View, locate the **Book a Flight: Mercury** row. (This is the row directly above the row displaying **passFirst0**.)

3. **Create a page checkpoint.**

   a. Right-click the **Book a Flight: Mercury** row, and select **Insert Standard Checkpoint**. The Page Checkpoint Properties dialog box opens.

When you run the test, UFT checks the number of links and images in the page, and the load time, against those described in the table at the top of the dialog box. The page load time may vary from computer to computer and your load time results may not match the above image.

UFT also checks the actual target URL of each link and the actual source of each image.

b. In the **Name** box, enter **CheckLinks** as the new checkpoint name.

c. Make sure that all three check boxes are selected in the **Type** column in the table at the top of the dialog box.

d. Accept the rest of the settings as default and click **OK**.

UFT adds the page checkpoint to your test. Because this is the first operation on the **Book a Flight: Mercury** page, the operation is added directly to the existing **Book a Flight: Mercury** page node. It is displayed in the Keyword View as a checkpoint operation on the **Book a Flight: Mercury** page.

| Book a Flight: Mercury | | | |
|---|---|---|---|
| Book a Flight: Me... | Check | CheckPoint("CheckLinks") | Check whether the "Book a Flight: Mercur |
| passFirst0 | Set | "Nicole" | Enter "Nicole" in the "passFirst0" edit box. |

4. **Save the test.**

Click **Save** 💾 .

Continue to enhance your test by adding a table checkpoint, as described in "Check Table Values" below.

# Check Table Values

In this exercise, you will add a table checkpoint to the test you created in "Create a Checkpoint Test" on page 81. The table checkpoint will check the cost of the outbound flight, as displayed in the **Book a Flight: Mercury** page.

1. **Start UFT and open the Checkpoint test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Click the **Open** button down arrow 📂 , and select **Open Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

   d. In the Solution Explorer, select the **Checkpoint** test node.

2. **Locate the step where you want to add a table checkpoint.**

   a. If the BookFlight action is not already open, in the Solution Explorer, double-click the **BookFlight** action. The action is displayed in the Keyword View.

b. Select the **passFirst0** step  (the step that enters the passenger's first name into the **First Name** edit box).

3. **Open the Mercury Tours Web site to the Book A Flight page.**

a. If it is not already open, log in to the Mercury Tours Web site: http://newtours.demoaut.com

b. In the **User Name** and **Password** boxes, enter `tutorial`.

c. Click **Sign-In**. The Flight Finder page opens.

d. Enter flight details.

Change the following selections by selecting from the drop-down lists. (Do not use the calendar option.)

- Departing From: **New York**

- On: **December 29**

- Arriving In: **San Francisco**

- Returning: **December 31**

> **Note:** If you are performing this tutorial in the month of December, or on the 29th or the 31st of any month, select a different month or date while recording. UFT records an operation only when you make a change in the application, so this ensures that the step is recorded. If you accept a default value (or re-select the default value), UFT does not record an operation.

e. Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

f. In the Select Flight page, click **CONTINUE** to accept the default flight selections. The Book a Flight page opens.

4. **Configure UFT to record on the open browser page.**

a. In UFT, select **Record > Record and Run Settings**. The Record and Run Settings dialog box opens.
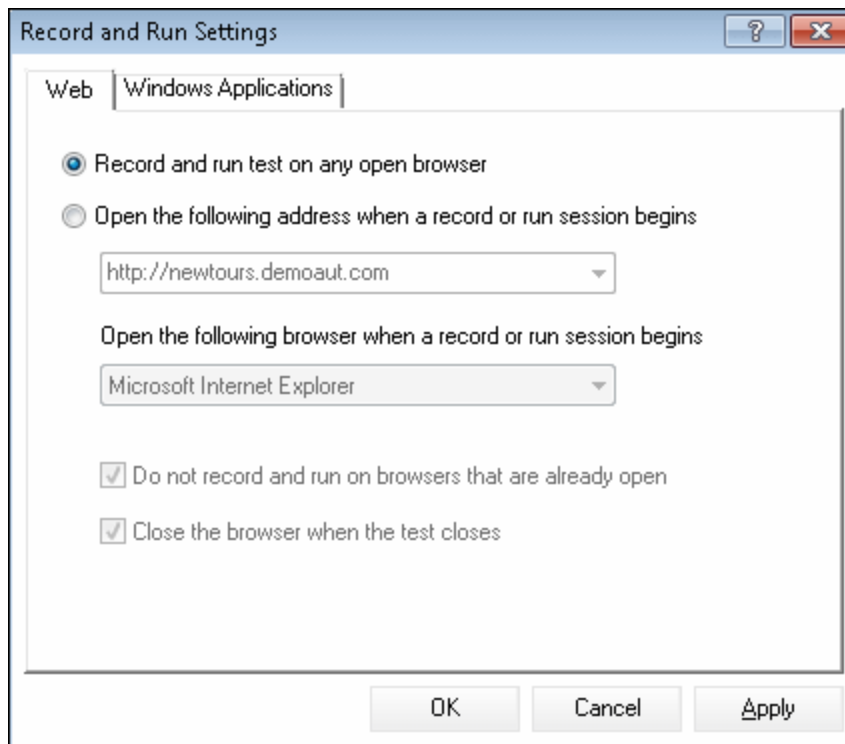
b. Select **Record and run test on any open browser** and click **OK** to close the dialog box.

5. **Create a table checkpoint.**

a. In the Book Flight page of the Mercury Tours Web site, under the flight departure date and the **Price** text, highlight **270** (the fare price of the first leg between New York and San Francisco).

b. In UFT, click the **Record** button ⬤. UFT begins a recording session and the UFT window is hidden.

c. Display the UFT window again and expand it if you need to.

d. Select **Design > Checkpoint > Standard Checkpoint**. The UFT window is hidden, and the pointer changes into a pointing hand.

> **Note:** If you need to switch back and forth between Welcome page, UFT, or any other window currently open, press **CTRL** to change the pointing hand back to a regular Windows pointer. Hold down the **CTRL** button as long as you need the pointer, and then release it when you are ready to use the pointing hand. Using the pointing hand to click objects that you do not need may cause errors in your test.

e.  Click the highlighted string, **270**. The Standard Checkpoint dialog box opens, displaying the object hierarchy.
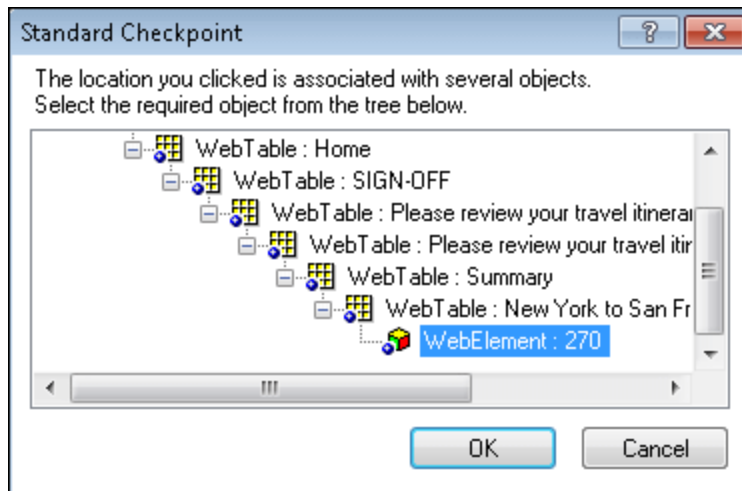


f.  Select the second-to-last item in the hierarchy (**WebTable: New York to San Francisco**), and click **OK**.

The Table Checkpoint Properties dialog box opens, showing the rows and columns of the table.

g.  Enter **CheckCost** as the new checkpoint name in the **Name** box of the Table Checkpoint
    Properties dialog box.
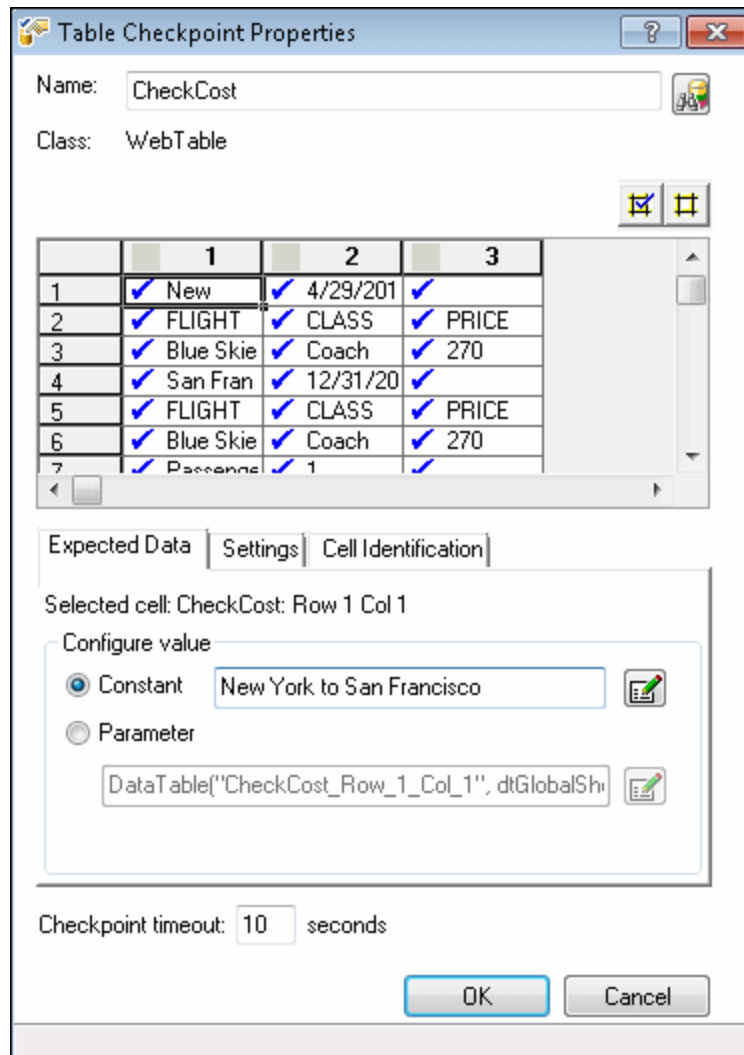


Note that by default, check marks appear in all cells. You can double-click a cell to toggle
the cell selection, or double-click a row or column header to toggle the selection for all the
cells in the selected row or column.

h.  Double-click each column header to clear the check marks.

i. Double-click column **3**, row **3** to check the value of this cell. (UFT checks only cells containing check marks.)



j. Scroll through the rows and columns to make sure that only the cell in column **3**, row **3** is checked. If any other cells are checked, double-click them to remove the check.

> **Tip:** You can change the column widths and row heights by dragging the boundaries of the column and row headers.

k. Accept the rest of the settings as default and click **OK**.

6. **Stop the recording session.**

Click **Stop**  to stop recording.

UFT adds the table checkpoint to your test. It is displayed in the Keyword View as a new step under the **Book a Flight: Mercury** page.



7. **Save the test.**

Click **Save**  .

In "Check Text Values" on the next page, you will need to navigate to the next page in the Mercury Tours Web site—the Flight Confirmation page. Therefore, you may want to keep the Mercury Tours Web site open to the current page.

Continue to enhance your test by adding a text checkpoint, as described in "Check Text Values" on the next page.

# Check Text Values

In this exercise, you will add a text checkpoint to the test you created in "Create a Checkpoint Test" on page 81. The text checkpoint will check whether **New York** is displayed in the **Flight Confirmation** page.

1. **Start UFT and open the Checkpoint test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. On the Start page, in the Recent Solutions area, click **Tutorial**. The Tutorial solution opens.

   c. In the Solution Explorer, select the **Checkpoint** test node. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

2. **Locate the page where you want to add a text checkpoint.**

   a. In the Solution Explorer, double-click the **BookFlight** action. The BookFlight action opens. If it opens in the Editor click the **Keyword View** button ▤ to display the Keyword View.

   b. In the Keyword View, highlight **Flight Confirmation: Mercury** step (located in the third to last row, if you have all the steps fully expanded).

3. **Open the Mercury Tours Web site to the Flight Confirmation page.**

   a. Log in to the Mercury Tours Web site: http://newtours.demoaut.com

      In the **User Name** and **Password** boxes, enter `tutorial`.

      Click **Sign-In**. The Flight Finder page opens.

   b. Enter flight details.

      Change the following selections by selecting from the drop-down lists. (Do not use the calendar option.)

      - Departing From: **New York**

      - On: **December 29**

      - Arriving In: **San Francisco**

      - Returning: **December 31**

> **Note:** If you are performing this tutorial in the month of December, or on the 29th or the 31st of any month, select a different month or date while recording. UFT records an operation only when you make a change in the application, so this ensures that the step is recorded. If you accept a default value (or re-select the default value), UFT does not record an operation.

   c. Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

   d. In the Select Flight page, click **CONTINUE** to accept the default flight selections.

   e. In the Book a Flight page, click **SECURE PURCHASE**. The Flight Confirmation page opens.

4. **Configure UFT to record on the open browser page.**

   a. In UFT, select **Record > Record and Run Settings**. The Record and Run Settings dialog box opens.
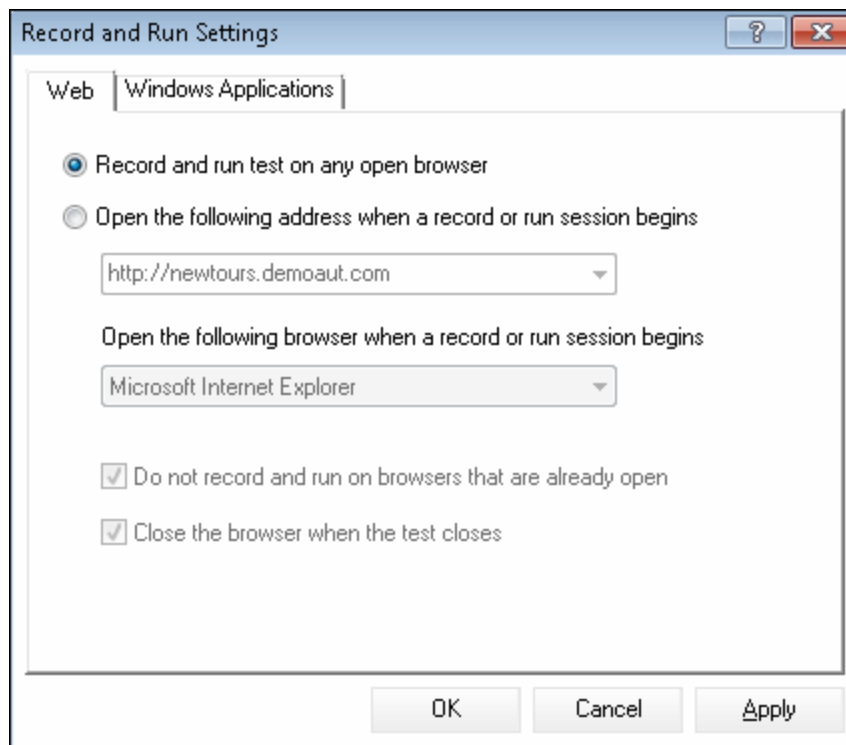


   b. Confirm that **Record and run test on any open browser** is selected and click **OK** to close the dialog box.

5. **Create a text checkpoint.**

    a. In the Flight Confirmation page, under **Departing,** highlight the text **New York** (include the space after it).

    b. If you need to, display and expand the UFT window.

    c. In the UFT window, click the **Record** button . UFT begins a recording session, and the Recording toolbar opens.

    d. Select **Design > Checkpoint > Text Checkpoint**.The UFT window is hidden, and the pointer changes into a pointing hand.

> **Note:** If you need to switch back and forth between Welcome page, UFT, or any other window currently open, press **CTRL** to change the pointing hand back to a regular Windows pointer. Hold down the **CTRL** button as long as you need the pointer, and then release it when you are ready to use the pointing hand. Using the pointing hand to click objects that you do not need may cause errors in your test.

e. Click the highlighted text string, **New York**. The Text Checkpoint Properties dialog box opens.



When **Checked Text** appears in the list box, the **Constant** field displays the text string you highlighted. This is the text UFT looks for when running the test.

f. In the **Name** box, enter CheckDepartingText as the new checkpoint name.

g. Remove **San Francisco** from the **Checkpoint Summary** pane, as follows:

○ Click **Configure** to open the Configure Text Selection dialog box.

○ Highlight the **to** text string. (Do not highlight **San Francisco**).

○ Click **Text After**. **San Francisco** changes from black to gray.
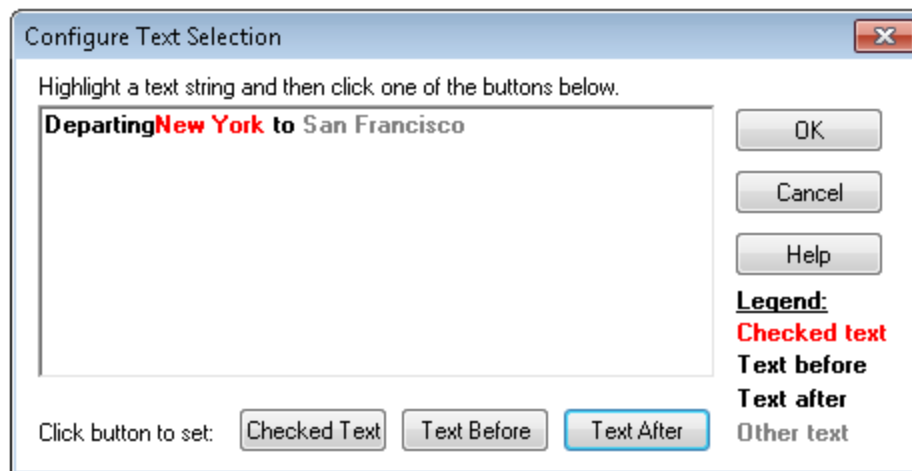


○ Click **OK** to close the dialog box. San Francisco is removed from the **text after** part of the Checkpoint Summary pane.
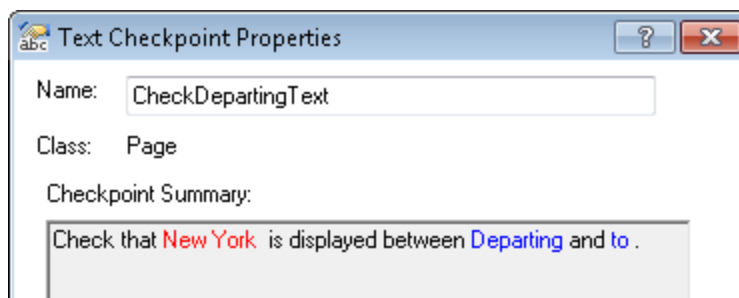


h. Accept the rest of the settings as default and click **OK**.

6. **Stop the recording session.**

On the Recording toolbar, click **Stop**  to stop recording.

UFT adds the text checkpoint to your test. It is displayed in the Keyword View as a checkpoint operation on the **Flight Confirmation: Mercury** page.

7. **Make sure that the checkpoint is located above the home step.**

Drag the new step up a row if needed, to make sure that UFT does not click **BACK TO HOME** before performing the checkpoint. In the Keyword View, the step should look as follows:

| | | | |
|---|---|---|---|
| ▼ ⑤ Flight Confirmation: Mercury | | | |
| ▼ ▪ Flight Confirmation: Mer... | Check | CheckPoint("CheckDepartingText") | Check whether text in the "F |
| ⊡ home | Click | | Click the "home" image. |

8. **Save the test.**

Click **Save** 💾 .

Learn more about checkpoint management in "Manage Checkpoints in the Object Repository" below, and then continue with "Run and Analyze a Test with Checkpoints" on page 102.

# Manage Checkpoints in the Object Repository

In addition to viewing checkpoints in a specific action, you can also view them in the object repository and modify their properties.

If applicable, you can even use the same checkpoint in more than one location. For example, if you want to verify that your organization's logo appears on every page in your application, you can create a checkpoint and insert it in wherever steps open another page.

For the purposes of this tutorial, you will not be reusing checkpoints.

1. **Start UFT and open the Checkpoint test.**

   a. If UFT is not currently open, open it as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Select **File > Open > Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

   > **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Open the Object Repository window.**

   a. In the Solution Explorer, double-click the **BookFlight** action. The BookFlight action opens.

   b. Click the **Object Repository** button 🏠 . The Object Repository window opens, displaying a tree of all test objects and all checkpoint and output objects in the current

action.

The tree includes all local objects and all objects in any shared object repositories associated with the action.
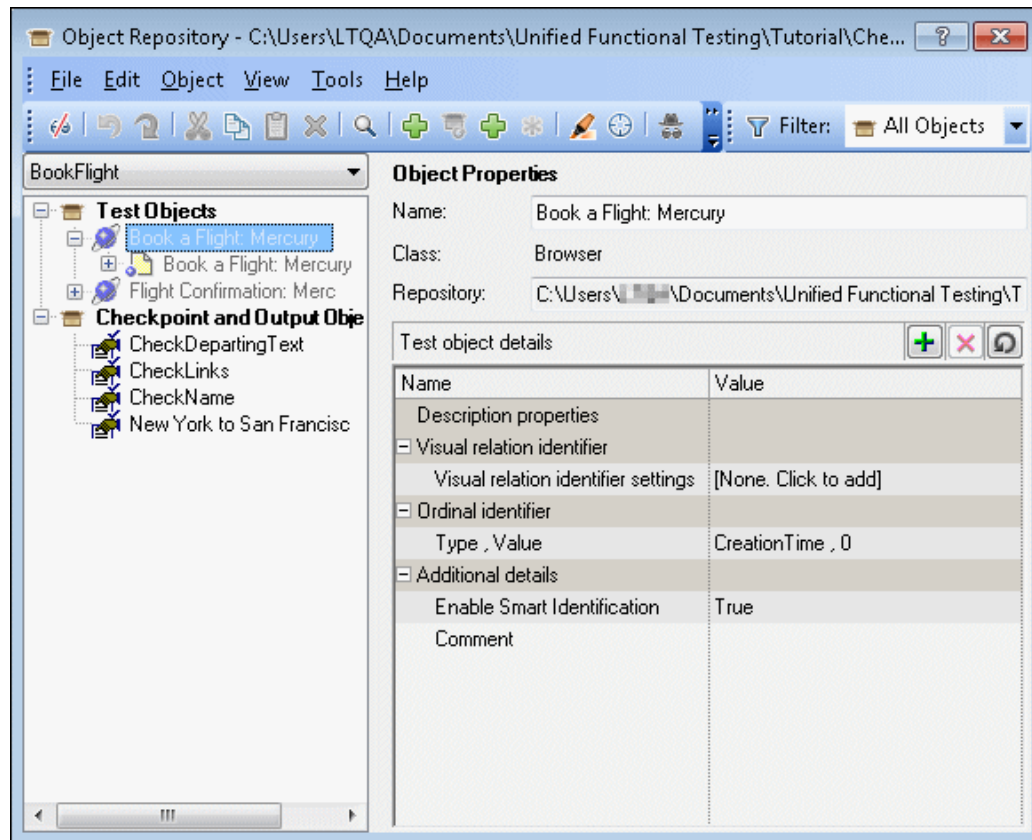


3. **Select an Action to view its checkpoints.**

    a. In the Action drop-down menu directly above the tree, select an action to display its test objects, checkpoint objects, and output value objects.

    b. Close the Object Repository Manager window when you are done.

> **Note:** For the purposes of this tutorial, you do not need to modify any object properties. For details about object properties, see the *HP Unified Functional Testing User Guide*.

You are ready to run your test. Continue with .

# Run and Analyze a Test with Checkpoints

In this exercise, you will run the test you created in "Create a Checkpoint Test" on page 81, and analyze the checkpoint results.

1. **Configure UFT to open the Mercury Tours Web page.**

    a. In UFT, select **Record > Record and Run Settings**. The Record and Run Settings dialog box opens.
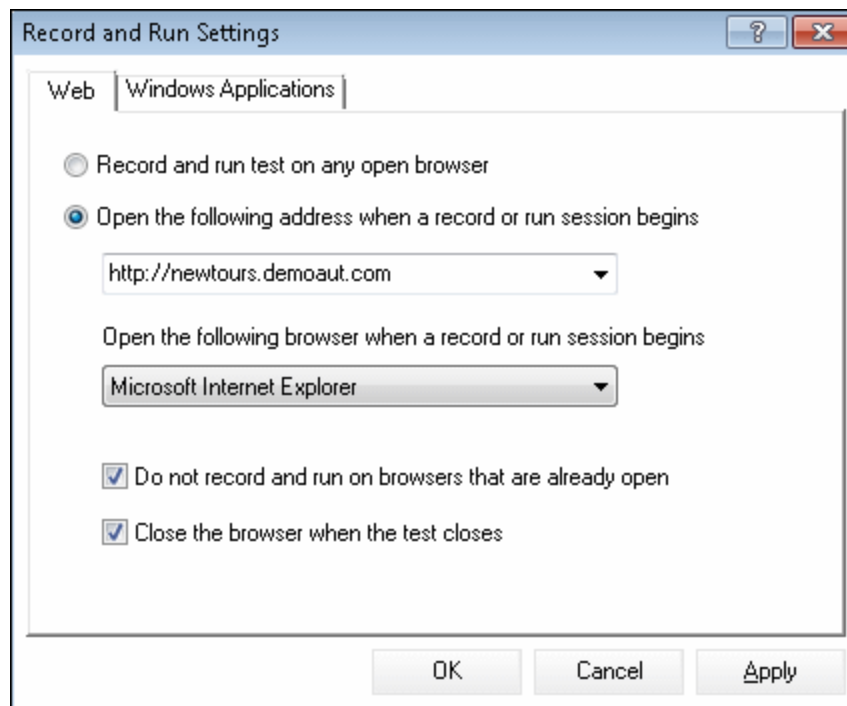


    b. Select **Open the following address when a record or run session begins** and click **OK** to close the dialog box.

2. **Start running your test.**

    a. Click the **Run** button [▷] . The Run dialog box opens.

    b. Ensure that **New run results folder** is selected. Accept the default results folder name.

    c. Click **OK**. When the test run is completed, the Run Results Viewer opens.

3. **View the run results.**

    When UFT finishes running the test, the Run Results Viewer opens. The run result should be **Passed**, indicating that all checkpoints passed. If one or more checkpoints had failed, the run

result would be **Failed**.

4. **View the results of the page checkpoint.**

   In the run results tree, expand all the child nodes of **Checkpoint Iteration 1 (Row 1)** until you reach **Book a Flight: Mercury**, and then select **Checkpoint** "**CheckLinks**".

   The Results Details pane displays all the details of the page checkpoint, including all items checked. You can drag the bottom of the Result Details pane to enlarge it and/or scroll within the pane to view all of the checkpoint details.

   > **Note:** The load time of the page may vary from computer to computer and your results may not match the image below.

The checkpoint passed because the actual values of the checked properties match the expected values.
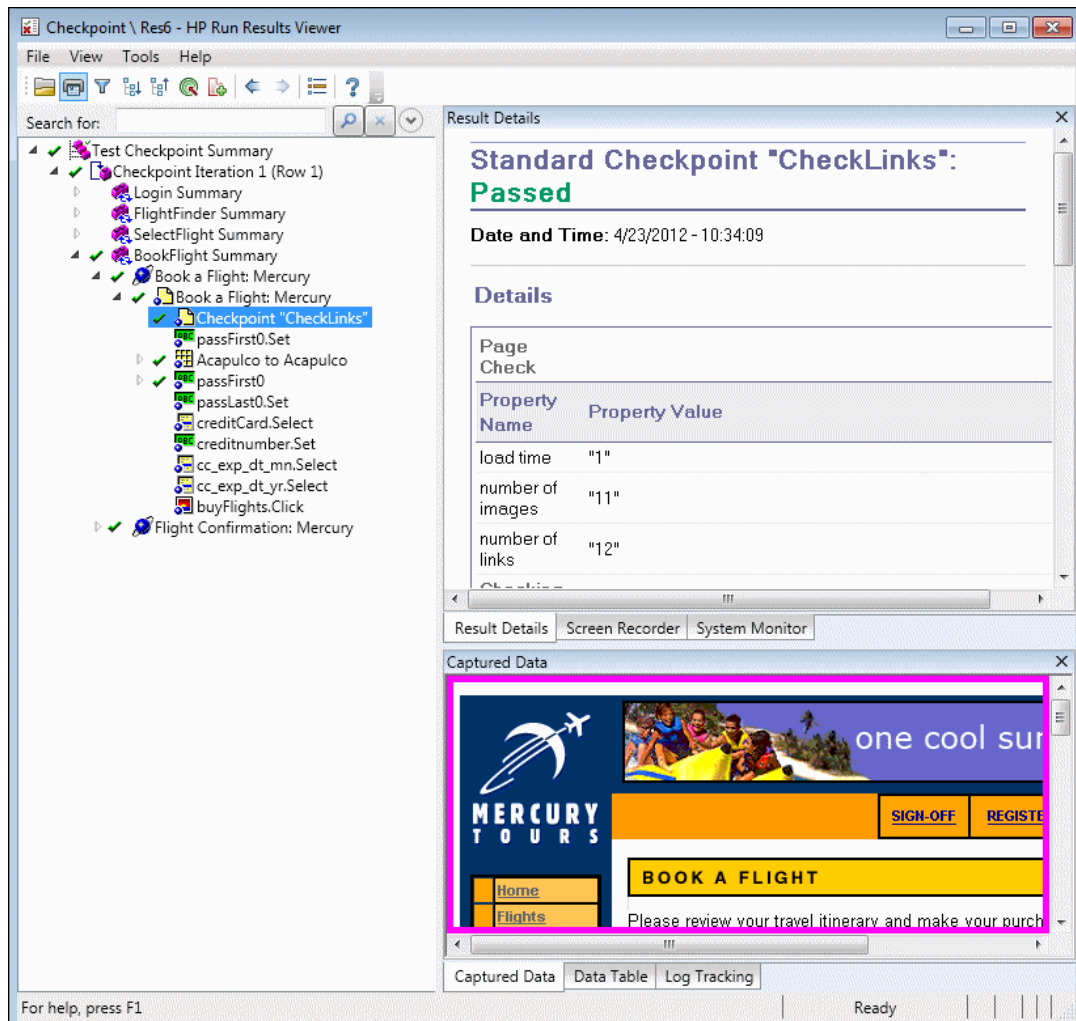


5. **View the results of the table checkpoint.**

   In the results tree, expand the **Book a Flight: Mercury>Acapulco to** node, and then select **Checkpoint "CheckCost"** .

   The Result Details pane displays the details of the table checkpoint, and the Captured Data pane displays the values of the table cells:

   ■ Cell values that were checked are displayed in black.

   ■ Cell values that were not checked are displayed in gray.

The checkpoint passed because the actual value of the checked cell matches the expected value.
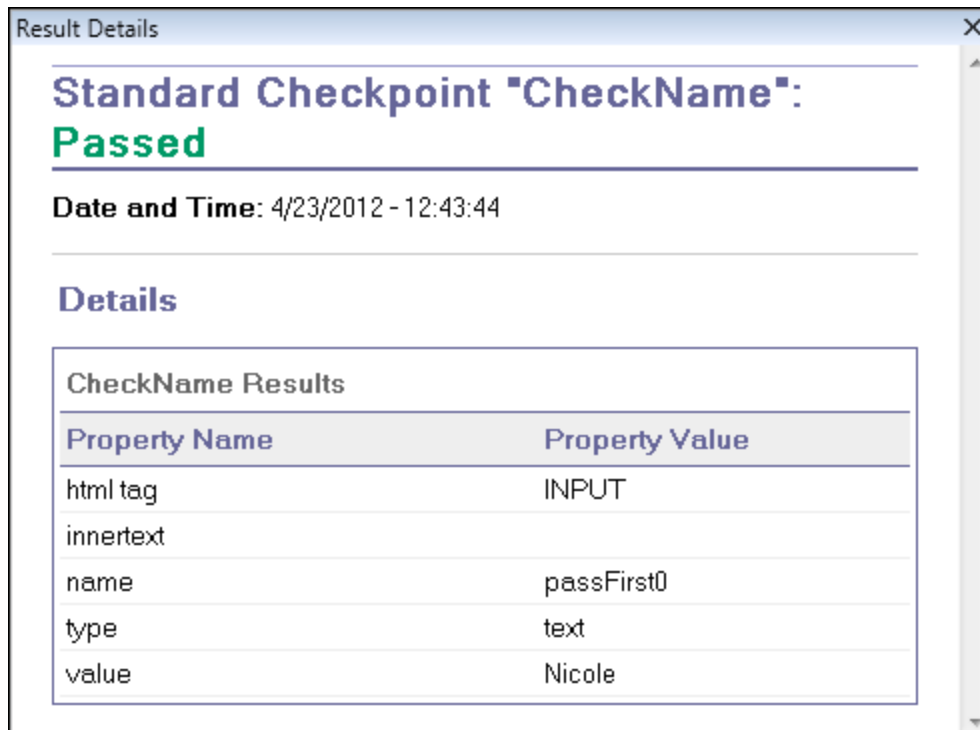


6. **View the results of the standard checkpoint.**

   In the results tree, expand the **Book a Flight: Mercury> passFirst0** node, and then select **Checkpoint "CheckName"**.

   The Results Details pane displays the details of the standard checkpoint, including the properties that were checked and their values.
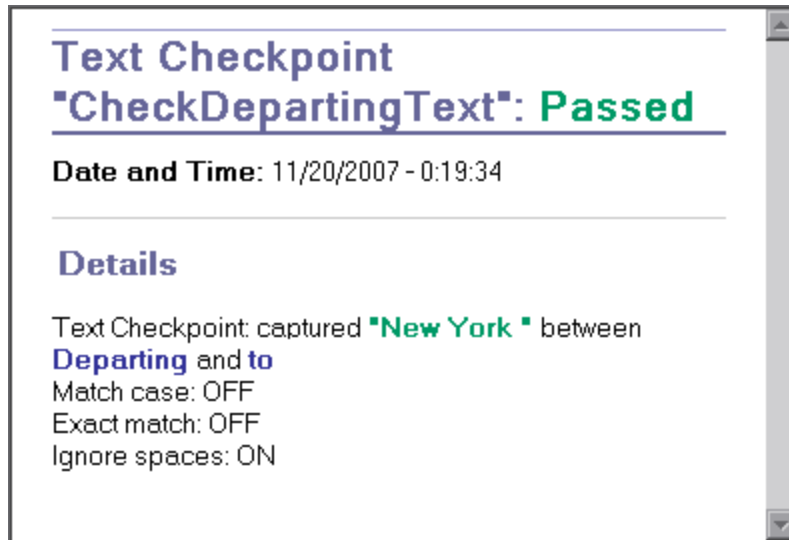
The checkpoint passed because the actual values of the checked properties match the expected values.

**Result Details** ✕

## Standard Checkpoint "CheckName": <span style="color:green">Passed</span>

**Date and Time:** 4/23/2012 - 12:43:44

## Details

### CheckName Results

| Property Name | Property Value |
|---|---|
| html tag | INPUT |
| innertext | |
| name | passFirst0 |
| type | text |
| value | Nicole |

7. **View the results of the text checkpoint.**

   In the results tree, expand **Flight Confirmation: Mercury** and highlight **Checkpoint "CheckDepartingText"**.

The Results Details pane displays the details of the text checkpoint. The checkpoint passed because the actual text matches the expected text.

**Text Checkpoint "CheckDepartingText": Passed**

**Date and Time:** 11/20/2007 - 0:19:34

**Details**

Text Checkpoint: captured **"New York "** between **Departing** and **to**
Match case: OFF
Exact match: OFF
Ignore spaces: ON

8. **Close the Run Results Viewer.**

   Select **File** > **Exit**.

You can also enhance your test using a function library. For details, continue with "Perform a Check Using a Function" below.

# Perform a Check Using a Function

In "Run and Analyze a Test with Checkpoints" on page 102, you used checkpoints to check various objects in the Mercury Tours Web site.

In this exercise, you will use the function you created in "Creating Functions and Function Libraries" on page 44 to check the date format of a Web element in the **SelectFlight** action.

1. **If UFT is not currently open, start UFT and open the Checkpoint test.**

   a. Open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. Click the **Open** button down arrow [icon], and select **Open Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

2. **Save the test as Function.**

   a. In the Solution Explorer, select the **Checkpoint** test node, and select **File > Save As**.

   b. In the Save Test As dialog box, browse to **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and save the test as `Function`.

   In the Solution Explorer, the Checkpoint test is replaced by the new Function test. The Checkpoint test is still saved separately in the file system.

3. **Add the Checkpoint test back to the solution.**

You can have the MercuryTours, Checkpoint, and Function test open at the same time, if they are all referenced from the same solution. This enables you to switch back and forth between them if you want to compare or edit the tests. You can only run a single test at a time.

   a. Click the **Add** button drop-down arrow [+ ▼] , and select **Add Existing Test**.

   b. Navigate to the **Checkpoint** test, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

   c. The Checkpoint test appears in the Solution Explorer. Tests are organized in the Solution Explorer alphabetically. The solution is saved automatically.

4. **Display the Select Flight page in The Mercury Tours Web site.**

   a. Log in to the Mercury Tours Web site: http://newtours.demoaut.com:

     ◦ In the **User Name** and **Password** boxes, enter `tutorial`.
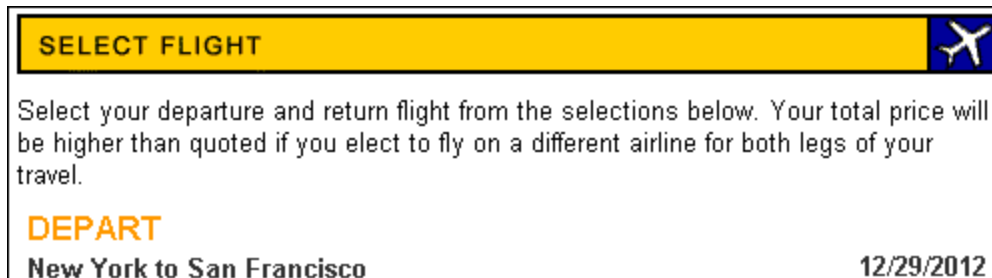
     ◦ Click **Sign-In**. The Flight Finder page opens.

b. Enter flight details.

Change the following selections by selecting from the drop-down lists. (Do not use the calendar option.)

- Departing From: **New York**

- On: **December 29**

- Arriving In: **San Francisco**

- Returning: **December 31**

> **Note:** If you are performing this tutorial in the month of December, or on the 29th or the 31st of any month, select a different month or date while recording. UFT records an operation only when you make a change in the application, so this ensures that the step is recorded. If you accept a default value (or re-select the default value), UFT does not record an operation.

c. Click **CONTINUE** to accept the other default selections. The Select Flight page opens. Notice the date displayed in the DEPART area. The date is in MM/DD/YYYY format.



5. **Add the departure date object to your repository.**

a. In the UFT window, select **Resources > Object Repository Manager**. The Object Repository Manager window opens.

b. In the Object Repository Manager, click **Open** 📁 , and browse to and open the **MercuryToursSelectFlight.tsr** object repository.

c. Click **Enable Editing** ✏️ to enable the repository for editing.

d. Click the browser window to make the Mercury Tours Web site active.

e. Learn the departure date object. This object is the departure date you selected when you entered your flight details.

In the sample image shown above, this date is **12/29/2012**.

○ From the Object Repository Manager, select **Object > Add Objects**. The Object Repository Manager window is minimized, and the cursor changes to a pointing hand.

> **Note:** If you need to switch back and forth between Welcome page, UFT, or any other window currently open, press **CTRL** to change the pointing hand back to a regular Windows pointer. Hold down the **CTRL** button as long as you need the pointer, and then release it when you are ready to use the pointing hand. Using the pointing hand to click objects that you do not need may cause errors in your test.

○ Click the departure date text string (in the example shown in this task, **12/29/2012**). The Object Selection - Add to Repository dialog box opens.
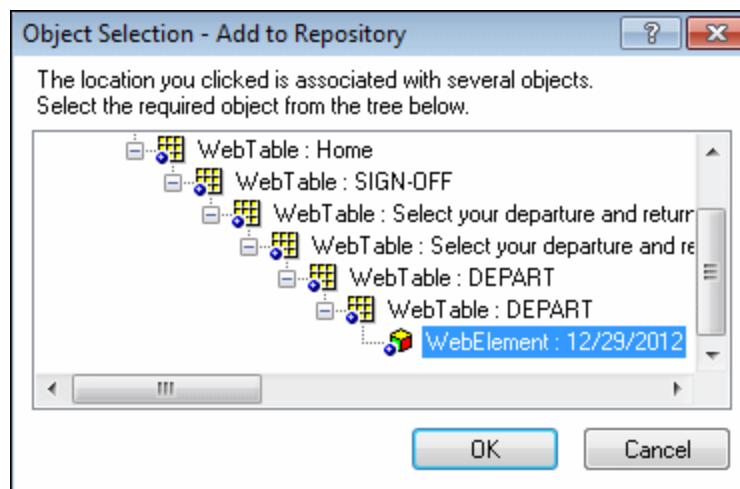


○ Make sure that the WebElement object that represents the departure date you selected when you entered your flight details is highlighted in the Object Selection dialog box.

○ Click **OK.** The departure date object is added to your repository.

f. Click **Save** 💾 to save the repository.

6. **In UFT, display the action in which you want to add a function.**

You want to add a function that checks the **innertext** property values of the date displayed in the **SelectFlight** action.

In the Solution Explorer, expand the **Function** test node, and double-click the **SelectFlight** action.

7. **Paste steps in the Editor.**

In this step, you will add the **check_data_validity** function defined in the associated function library to the **SelectFlight** action. You created this function library in "Create a Function" on page 45.

a. Click the **Editor** button to display the Editor

b. In the Editor, place your cursor *before* the single existing step, and paste the function shown below.

```
departureDate=Browser("Select a Flight: Mercury").Page("Select a
Flight: Mercury").WebElement("12/29/2012").GetROProperty("innertext")

if check_data_validity(  departureDate ) then
            reporter.ReportEvent micPass, "Date is valid" , departureDate
end if
```

c. After you paste the steps, make sure that the original step still starts on its own line after the pasted steps. If it does not, place your cursor before the step starts, at **Browser ("Select a Flight: Mercury")**, and press Enter.

You want the function to run *before* UFT navigates to the next page in the application.

d. In the first line of the function, modify the date to the date you selected when you entered your flight details, which will match the departure date object you added to the object repository.

e. In the PDF version of this tutorial, the first line of the code breaks across two lines. If you are copying and pasting from the PDF, make sure to remove the line break after **29/:** so that the first line of the code begins with **departureDate** and ends with **("innertext")**. (The **Page** part should be displayed as: Page("Select a Flight: Mercury") with a single space between **Flight:** and **Mercury**.)

8. **Look at these steps in the Keyword View.**

Select **View > Keyword** to display the Keyword View. (The highlighted step is not part of the function—it is the step that navigates to the next page in the Mercury Tours Web site.)

| Item | Operation | Value | Documentation |
|---|---|---|---|
| ▼ SelectFlight | | | |
| ▼ Select a Flight: Mercury | | | |
| ▼ Select a Flight: Mercury | | | |
| 12/29/2012 | GetROProperty | "innertext" | Retrieve the current value of the "innert |
| ▶ IF Function Call | check_data_vali... | departureDate | |
| reserveFlights | Click | | Click the "reserveFlights" image. |

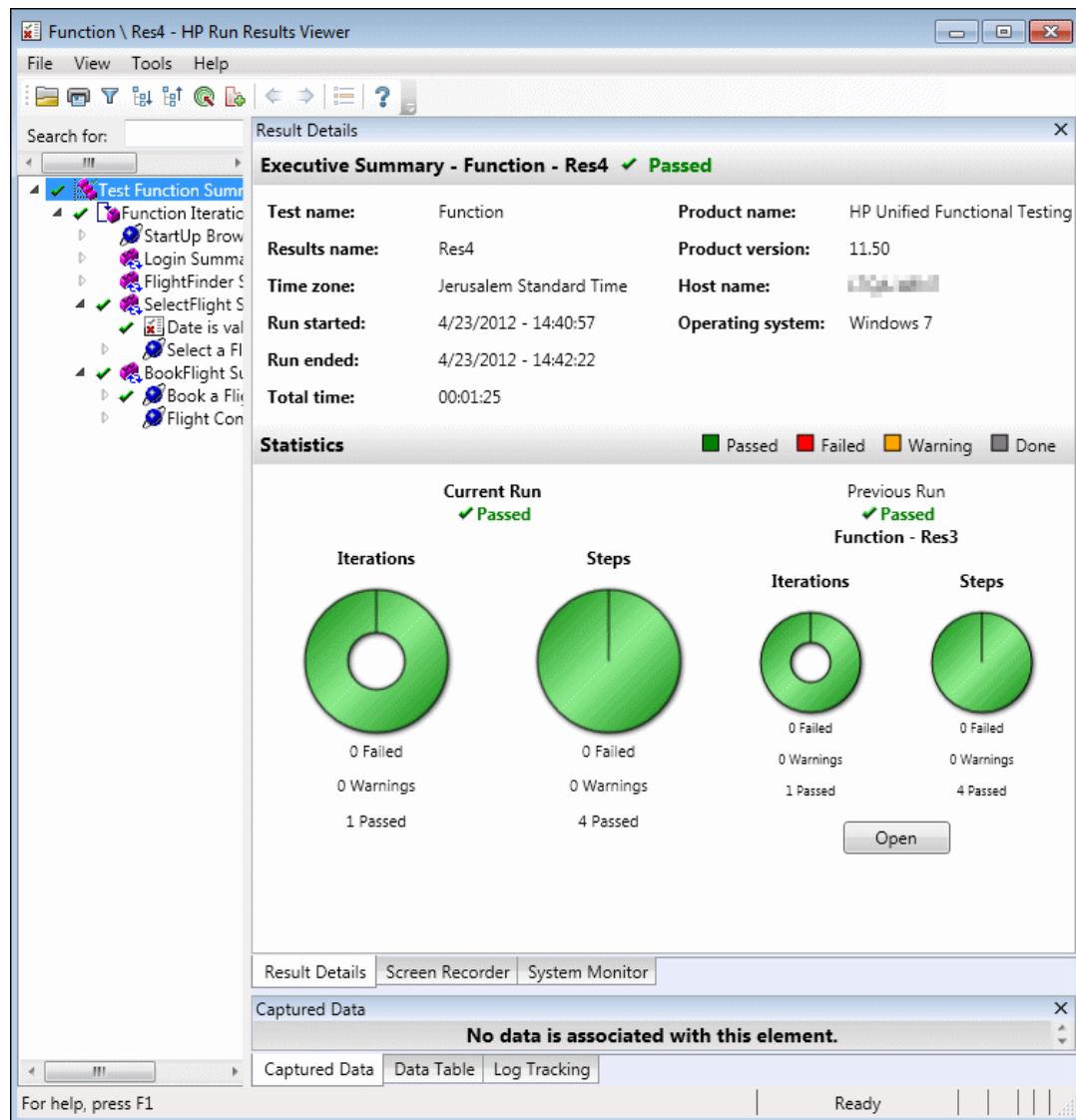9. **Save your test.**

   Select **File > Save**.

10. **Start running your test.**

    a. Click the **Run** button [▷] . The Run dialog box opens.

    b. In the Run dialog box, ensure that **New run results folder** is selected. Accept the default results folder name.

    c. Click **OK**. When the test run is completed, the Run Results Viewer opens.

11. **Analyze the run results.**

    The run result should be **Passed**, indicating that the function passed because the date is displayed in the expected format, and it contains acceptable values. For details about the Run Results Viewer, see "Navigating the Run Results Viewer" on page 76 and the *HP Run Results Viewer User Guide*.

12. **Close the Run Results Viewer.**

Select **File > Exit** to close the Run Results Viewer.

Continue to enhance your test by adding multiple sets of data, as described in "Parameterizing Steps and Objects" on page 114.

# Chapter 8: Parameterizing Steps and Objects

In "Running and Analyzing Tests" on page 73, you ran a test that you created in previous lessons, to check that a series of steps performed on the Mercury Tours Web site runs smoothly. Even after adding additional checks in "Creating Checkpoints and Using Functions" on page 79, you ran this test with only a single set of data. However, when you test *your* applications, you may want to see the same operations performed with multiple sets of data.

For example, you may want to run a test on your Web site using ten separate sets of data. You can create ten separate tests, each with its own set of data, or you can add ten sets of parameters to a single test. If you add the parameters, your test will run ten times, each time using a different set of data.

In this lesson, you will add parameters to your test and run your test with multiple sets of data.

This chapter includes:

# Parameterizing an Action

The Data pane at the bottom of the UFT window in the any test displays a tab for each action in the test, as well as a **Global** tab.

> **Note:** If the Data pane is not displayed, select **View > Data** to display it, or click the **Data** button ⊞⊞ .

The Global tab is a data sheet whose data is used for the entire test. If five rows of data are displayed in the Global data table, the test will run five times (five full iterations). In addition, you can create data sets for each action, using the relevant action sheet. If you parameterize a step using a Data pane parameter from a local action sheet and enter five rows of data in that sheet, you can define that action to run five times within one test iteration.

UFT enables you to insert various types of parameters. This tutorial introduces you to Global Data pane parameters. For details on other types of parameters, see the *HP Unified Functional Testing User Guide*.

# Create a Test for Parameterization

In "Adding Steps " on page 50, you reserved a flight from New York to San Francisco. In those steps, New York is a constant value, which means that New York is the departure city each time you run the test. In this exercise you will create a new test, in which you define the departure city as a parameter, so that you can use a different departure city for each test run.

1. **Start UFT and open the Checkpoint test.**

   a. Open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. On the Start page, in the Recent Solutions area, click **Tutorial**. The Tutorial solution opens.

      The Checkpoint, Function, and MercuryTours tests appear in the Solution Explorer.

2. **Save the test as Parameter.**

   a. In the Solution Explorer, select the **Checkpoint** test node. (You created the Checkpoint test in "Create a Checkpoint Test" on page 81.)

   b. Select **File > Save As**. In the Save Test As dialog box, browse to **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and save the test as `Parameter`.

      In the Solution Explorer, the Checkpoint test is replaced by the new Parameter test. The Checkpoint test is still saved separately in the file system.

3. **Add the Checkpoint test back to the solution.**

You can all of your tests open at the same time if they are both referenced from the same solution. This enables you switch back and forth between them if you want to compare or edit the tests. You can only run a single test at a time.

a. Select **File > Add > Existing Test**.

b. Navigate to the **Checkpoint** test located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Add**. (You created this test in "Create a Checkpoint Test" on page 81.)

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

c. The Checkpoint test appears in the Solution Explorer. Tests are organized in the Solution Explorer alphabetically.

The solution is saved automatically. Continue with "Define a Data Table Parameter" below.

# Define a Data Table Parameter

In this exercise you will define the departure city as a parameter, so that you can use a different departure city for each test run.

1. **Start UFT and open the Parameter test.**

a. Open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

b. Select **File > Open > Solution**. The Open Solution dialog box opens.

c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Parameter test in "Create a Test for Parameterization" on the previous page.)

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

d. In the Solution Explorer, select the **Parameter** test node.

2. **Make sure that the Data Pane option is visible.**

If you do not see the Data pane at the bottom of the UFT window, select **View > Data**.
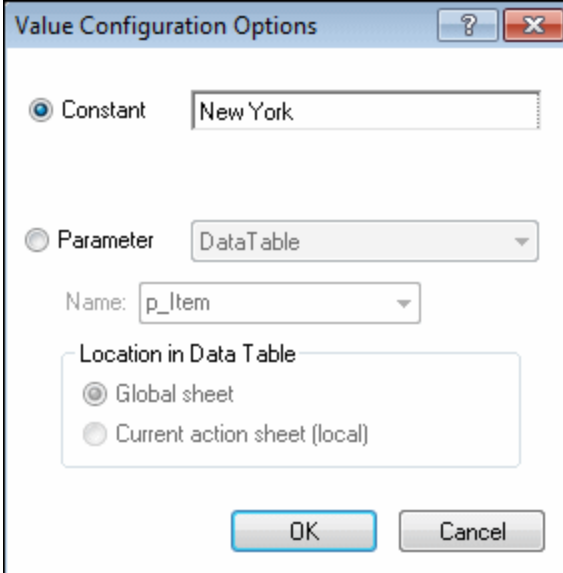
3. **Open the FlightFinder action.**

   In the canvas, double-click the FlightFinder action. The FlightFinder action opens. If the Editor is displayed, select **View > Keyword View** to display the Keyword View.

4. **Select the text to parameterize.**

   In the Keyword View, in the **fromPort** row, click the **Value** cell and then click the parameterization button .

   The Value Configuration Options dialog box opens.



5. **Set the parameterization properties.**

   a. Select the **Parameter** radio button. This enables you to replace the constant value (**New York**) with a parameter.

b. Confirm that the **DataTable** option is selected from the Parameter drop-down menu. This means that the value of the parameter will be taken from the UFT Data pane. The **Name** box is enabled and displays **p_Item**. Delete **p_Item** and enter departure.
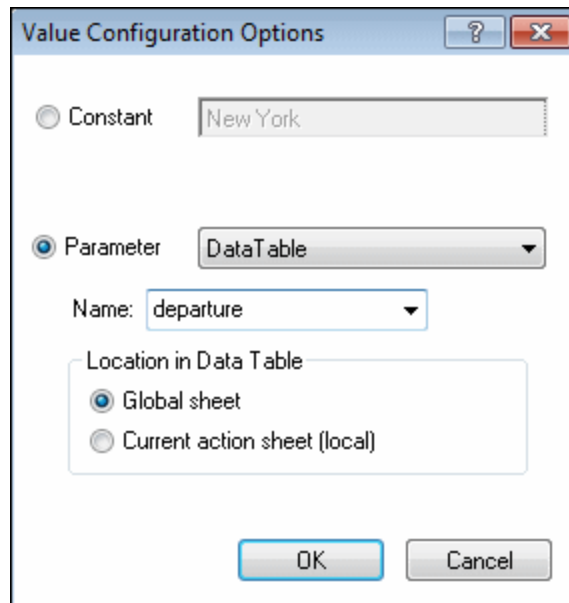


c. Click **OK** to close the dialog box. UFT adds the **departure** parameter to the Data pane as a new column and inserts New York in the first row under it.

New York will be the first of several departure cities that UFT will use during test runs of the application.
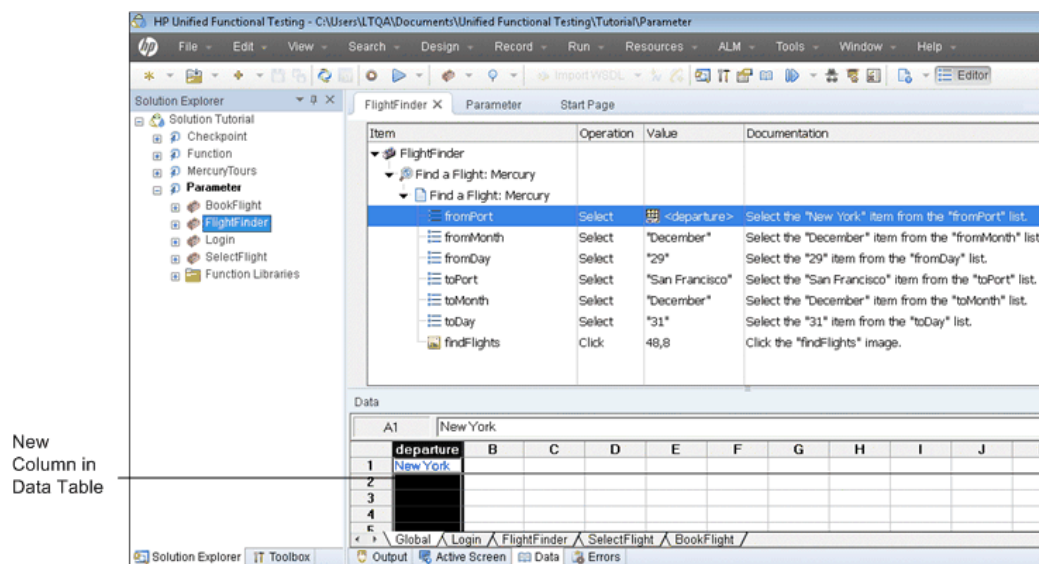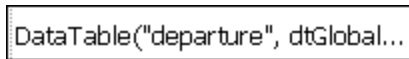


Note the change in the step's appearance in the Keyword View. Previously, the step was displayed as fromPort Select New York. Now, when you click the **Value** cell, the

following information is displayed, indicating that the value is parameterized using a Data pane parameter called departure:



When you click another area in the Keyword view, the step is displayed as:



Continue with .

# Add Parameter Values to a Data Table

As you saw in , UFT displays parameter values in the Data pane. In this exercise, you will add another departure city to the Data pane (in the FlightFinder action of the Parameter test), so that UFT can test the application with this data.

1. **Start UFT and open the Parameter test.**

   a. Open UFT as described in , making sure that only the Web Add-in is loaded.

   b. Click the **Open** button down arrow , and select **Open Solution**. The Open Solution dialog box opens.

   c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. (You created the Parameter test in .)

      **Note:** If you copy this path, make sure to replace **<username>** with your username.

   d. In the Solution Explorer, select the **Parameter** test node.

2. **Open the FlightFinder action.**

   In the Solution Explorer, double-click the **FlightFinder** action.

3. **Enter an additional city in the departure column.**

   In the Data pane, click row **2** in the **departure** column, enter London and then press **Enter**. (If the Data pane is not displayed, select **View > Data**.)

4. **Save the test.**

   Click **Save** .

# Modify Steps Affected by Parameterization

After parameterizing one step in a test, test objects in other steps might be affected when the value of the parameterized step changes. If so, you must modify the expected values of those objects to match the value resulting from the parameterized step.

In "Add Parameter Values to a Data Table" on the previous page, you added parameter values for the departure city in the FlightFinder action. In this exercise, you will modify the text checkpoint so that UFT checks for the text that matches the current departure city during the test run.

1. **Start UFT and open the Parameter test.**

   a. Open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

   b. On the Start page, in the Recent Solutions area, click **Tutorial**. The Tutorial solution opens, displaying the MercuryTours, Checkpoint, and Parameter tests.

   c. In the Solution Explorer, select the **Parameter** test node. (You created the Parameter test in "Create a Test for Parameterization" on page 115.)

2. **Locate the text checkpoint to modify.**

   a. In the Solution Explorer, double-click the Parameter test **BookFlight** action. The

      BookFlight action opens. (If the Editor is displayed, click the **Keyword View** button to display the Keyword View.)

   b. Right-click the **Flight Confirmation: Mercury** row with the existing checkpoint and select **Checkpoint Properties**.

      > **Note:** You may need to expand the **Flight Confirmation: Mercury** step to see the row with the checkpoint.

The Text Checkpoint Properties dialog box opens. In the **Checked Text** area, **New York** is displayed in the **Constant** box. **New York** is the expected value for the checkpoint for every iteration.
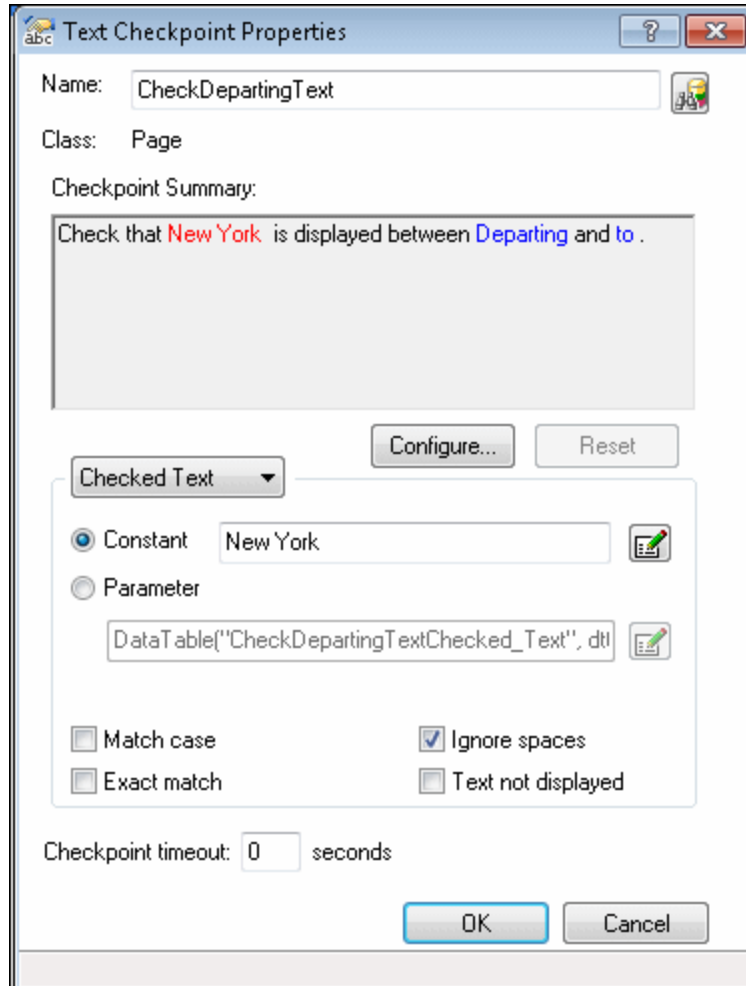


3. **Parameterize the text checkpoint.**

   a. Select **Parameter** and click the **Parameter Options** button . The Parameter Options dialog box opens.
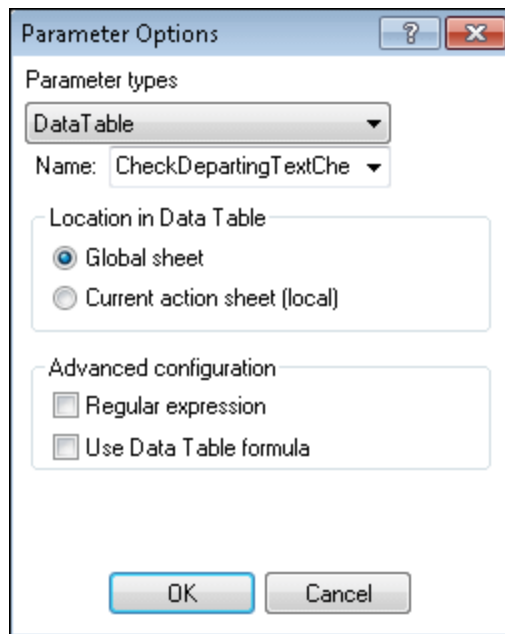
b. In the **Name** box, select **departure**. This instructs the checkpoint to use the departure parameter values in the Data pane for the expected results.

c. Click **OK** to close the Parameter Options dialog box, and then click **OK** again to close the Text Checkpoint Properties dialog box. The checkpoint is now parameterized.

4. **Save the test.**

Select **File > Save** or click **Save** 💾 .

You can now run your test. Continue with "Running and Analyzing a Parameterized Test" below.

# Running and Analyzing a Parameterized Test

In "Modify Steps Affected by Parameterization" on page 120, you finished creating a single data set for the Parameter test.

In this exercise, you will run the modified Parameter test. UFT will run the test twice, once for a departure from New York, and a second time for a departure from London. If you go back and add additional parameters in the Data pane (such as additional departure cities), additional iterations will be added for each data set, each represented by a row in the data table.

1. **Start UFT and open the Parameter test.**

a. Open UFT as described in "Create a New Solution, Test, and Actions" on page 25, making sure that only the Web Add-in is loaded.

b. Select **File > Open > Solution**. The Open Solution dialog box opens.

c. Navigate to the **Tutorial.ftsln** file, located in **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial**, and click **Open**. The Tutorial solution opens, displaying the MercuryTours, Checkpoint, and Parameter tests.

> **Note:** If you copy this path, make sure to replace **<username>** with your username.

d. In the Solution Explorer, select the **Parameter** test node. (You created the Parameter test in "Create a Test for Parameterization" on page 115.)

2. **Configure UFT to open the Mercury Tours Web page.**

a. In UFT, select **Record > Record and Run Settings**. The Record and Run Settings dialog box opens.
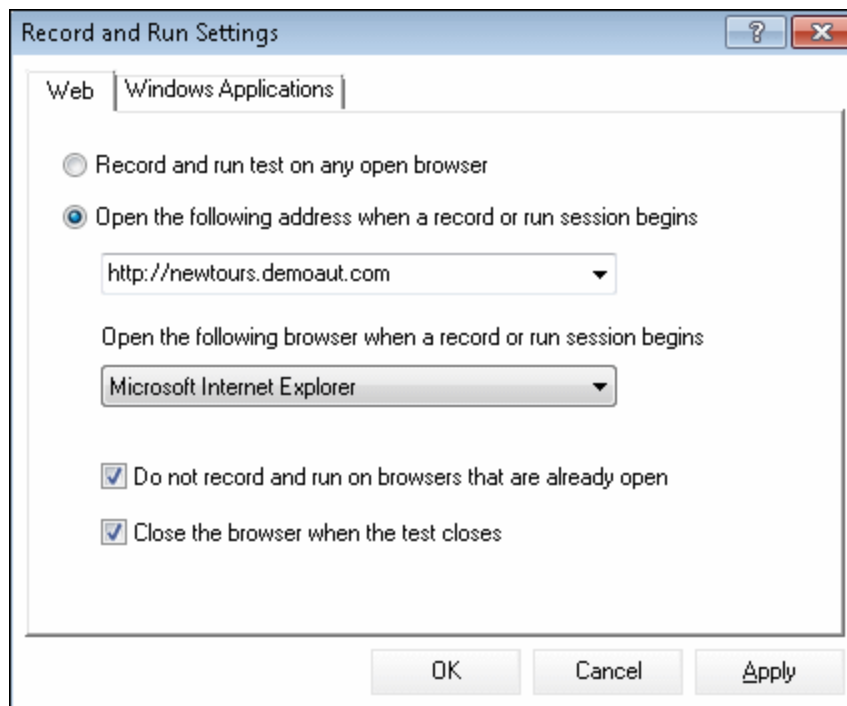


b. Select **Open the following address when a record or run session begins** and click **OK** to close the dialog box.

3. **Run the Parameter test.**

a. Click the **Run** button . The Run dialog box opens.

b. Select **New run results folder** and accept the default results folder name.

c. Click **OK**. When the test run is completed, the Run Results Viewer opens.

4. **Examine the results summary.**
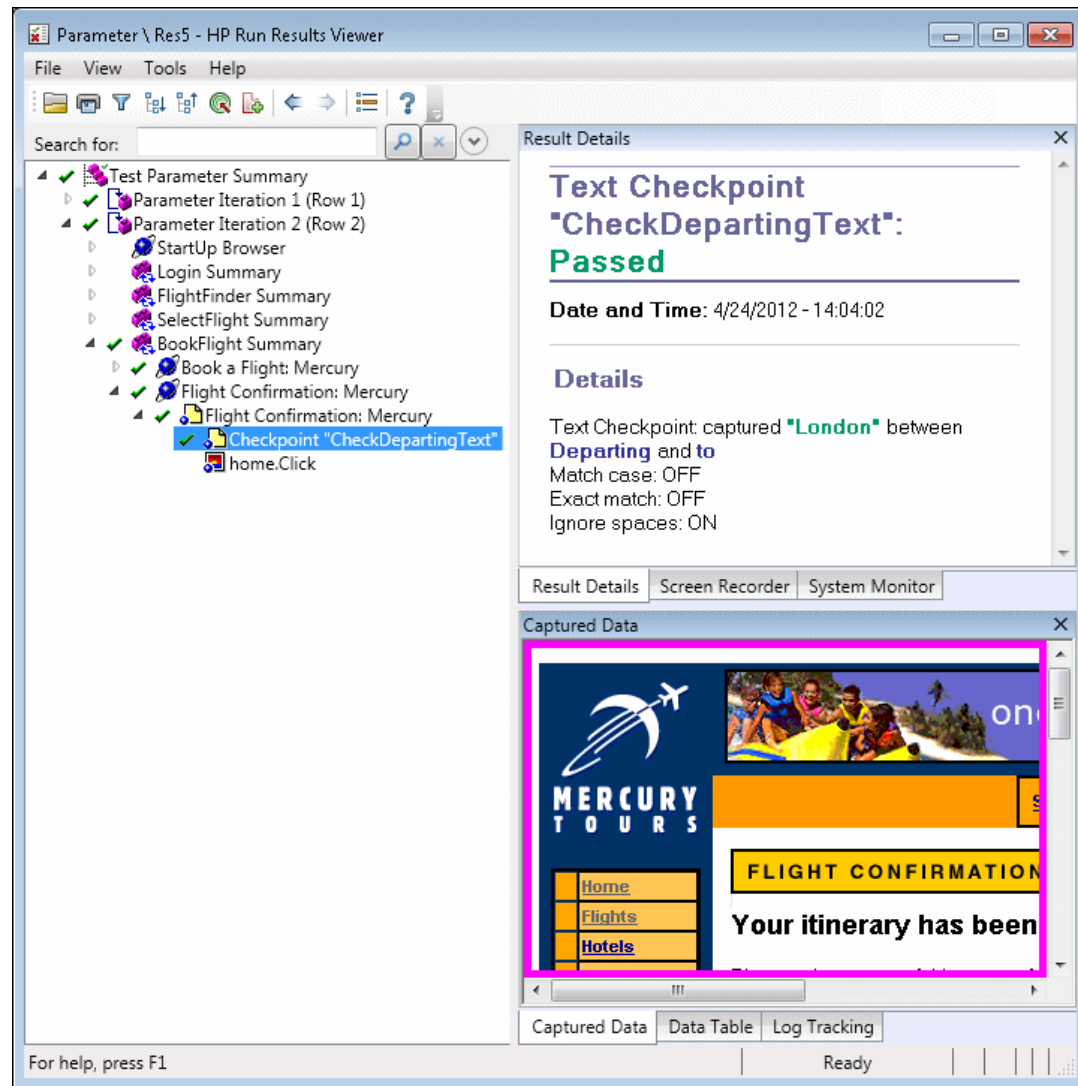
   The Run Results Viewer shows that the iterations of the test passed. Drill down in the run results tree to view the various results.



5. **Close the Run Results Viewer.**

   Select **File > Exit** to close the Run Results Viewer.

You've now completed the exercises in this tutorial. Continue with applying the concepts and skills you learned to testing your own applications, as described in "Where Do You Go From Here?" on page 125.

# Chapter 9: Where Do You Go From Here?

In "Parameterizing Steps and Objects" on page 114, you completed adding all the test enhancements described in this tutorial. You are now ready to apply the UFT concepts and the skills you learned to testing your own application.

This chapter includes:

# Modifying UFT Settings to Save Disk Space

Some of the settings you configured during this tutorial require a lot of disk space. Therefore, you may want to modify these settings to save disk space.

In "Run a Test" on page 74, you configured UFT to save all images to the run results. Now, you will configure UFT to save images only when a step fails.

1. **Start UFT.**

   Open UFT as described in "Create a New Solution, Test, and Actions" on page 25. There are no add-in requirements in this lesson.

2. **Modify the global settings for saving images to the run results.**

   a. Select **Tools > Options > GUI Testing** tab **> Screen Capture** node.

   b. In the **Save still image captures to results** option, select **For errors**. (You can also clear the **Save still image captures to results** check box so that no images are ever saved to the run results. However, this may make troubleshooting your tests more difficult.)

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

# Testing Your Own Application - Getting Started

This tutorial covers the basic tools needed for testing applications and Web sites. We suggest that you use the following procedure when testing your own application.

1. **Analyze your application.**

   - Determine the development environment. This enables you to load the relevant UFT add-ins and provide support for the objects in your application.

   - Determine the business processes that users will perform. Plan your tests and actions accordingly.

   - Decide how to organize your test and which operations to include. Consider the goals of the test, and confirm that your application and UFT are set to match the needs of your test.

   At this stage, you can begin creating the skeletal tests and actions to use when testing your application.

2. **Prepare your testing infrastructure.**

   Decide how to store the objects in your test. You can store the objects for each action in its corresponding local object repository, or you can store the objects for each action in one or

more common (shared) object repositories. You can also use the same shared object repository for multiple actions.

- **If you are new to testing**, you may want to use a local object repository for each action. This is the default setting, and all objects are automatically added to the local repository of each action.

- **If you are familiar with testing**, it is often most efficient to work with shared object repositories, which can be used for one or more actions. Object information is kept in one central location, and when the objects in your application change, you can update them in that one location for multiple actions, in multiple tests.

Although not discussed in this tutorial, you can also export test objects from a local object repository to a shared object repository, and you can merge object repositories.

You may also want to create function libraries to enhance UFT functionality.

For details, see the *HP Unified Functional Testing User Guide*.

3. **Build your test.**

   While you create your test steps (described in "Adding Steps " on page 50), follow the steps you expect users to perform as they navigate within your application.

4. **Enhance your test.**

   - Add checkpoints (described in "Creating Checkpoints and Using Functions" on page 79) to search for specific values of a page, object, text string, or table cell.

   - Replace fixed values in your test with parameters (described in "Parameterizing Steps and Objects" on page 114) to check how your application performs the same operations with multiple sets of data.

   You can further enhance your test with programming, conditional, and loop statements, which add logic to your test. For details, see the *HP Unified Functional Testing User Guide*.

5. **Debug your test.**

   Debug your test to check that it operates smoothly and without interruption. For details, see the *HP Unified Functional Testing User Guide*.

6. **Run your test.**

   Run your test on your application (described in "Running and Analyzing Tests" on page 73) to check that the application functions as expected.

7. **Analyze the run results.**

   Examine the results of your test to pinpoint defects in your application. (Refer to the appropriate sections of this tutorial to understand what to look for in the run results for checkpoints or parameters.)

8. **Report defects.**

   If you have ALM installed, you can submit any defects discovered to an ALM database. (ALM is the HP solution for test management. For details, see the *HP Application Lifecycle Management User Guide* and any other documentation included with ALM.)

# We appreciate your feedback!

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on GUI Testing Tutorial (Unified Functional Testing 12.01)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to sw-doc@hp.com.