Creating Data-Driven Tests Introduction

Time Required

To complete this tutorial, you will need approximately **90 minutes**.

About the Sample Application -- Classics CD

This tutorial uses a sample application to demonstrate how to data-drive a test. The Classics CD application that comes with Rational Functional Test lists classical music CDs available for purchase. Customers can purchase a Classical music CD and charge their purchases to a credit card.

Learning Objectives

This tutorial is divided into eight exercises that must be completed in sequence for the tutorial to work properly. In this tutorial, you will use the ClassicsCD sample application to create a project and record a test script to verify that the ClassicsCD sample application correctly totals an order placed in the application. You will data-drive a test so that you can use a variety of realistic data to test the application. You will also create a verification point with a datapool reference to check that the total amount of the order is correct in the Classics CD application.

While completing the exercises, you will:

- Create a project and record a test script
- Data-drive a test
- Add descriptive headings to the data
- Create a verification point with a datapool reference
- Add data to the datapool
- Play back the test

When you are ready, begin Exercise 1.1: Creating a Project and Recording a Test Script.

Exercise 1.1: Creating a Project and Recording a Test Script

In this exercise, you will use the Classics CD sample application to create a new project and start recording a test to verify that the sample application correctly totals the amount of music CDs purchased.

A project is a collection of test assets such as test scripts, object maps, verification points, and datapools, that can facilitate the testing of one or more software components. You must create a Functional Test project before you can record a test.

Create a Project

Create a project to store the test assets you need to test the Classics CD sample application.

1. Start Functional Test.
2. Click **File > New > Functional Test Project**.
3. Type **DataDriveTutorial** for the name of the new project.
4. Click **Finish**.

Start Recording

Start recording a test script to verify that when a customer orders a music CD, the total amount charged to the credit card is the correct amount listed in the application.

1. On the Functional Test toolbar, click **Record a Functional Test Script** ( ).
2. Type **OrderTotal** for the name of the test script.
3. Click **Next**.

   The Select Script Assets page opens.

   When you create a test script, Functional Test creates a test datapool and other test assets. For this tutorial, use the defaults on this page: **Private Test Datapool** and **Sequential**. A private test datapool is associated with only one script and is not shared by any other scripts. When you use the sequential order, the test script accesses records in the datapool in the order that they appear in the datapool.

4. Click **Finish**.

   The Functional Test window minimizes and the Recording Monitor opens.

Start the ClassicsCD Application

Start the ClassicsCD application and navigate through the application to the dialog box that you will data-drive.

1. On the **Recording** toolbar, click **Start Application** ( ) to start an application.

2. If necessary, click the **Application Name** arrow, to see the options, and then select **ClassicsJavaA - java**.
3. Click **OK**.

   ClassicsJavaA is build 1 of the sample application, ClassicsCD, which comes with Functional Test.

   The sample application starts.

4. In the ClassicsCD application, under **Composers**, double-click **Schubert** to open the CDs for sale by that composer, and then click **String Quartets Nos. 4 & 14**.
5. Click **Place Order**.
6. Click **OK** to close the Member Logon dialog box.

   The Place an Order dialog box opens.

7. In the ClassicsCD application, type 1234567890 in the **Card Number** box and 09/09 in the **Expiration Date** box.

Now you are ready to begin Exercise 1.2: Data-driving a Test.

Exercise 1.2: Data-driving a Test

Before you begin, you must complete Exercise 1.1: Creating a Project and Recording a Test Script.

In this exercise, you will use the data-driver to populate a datapool with data from the sample application. A datapool is a collection of related data records. A datapool supplies data values to the variables in a test script during test script playback.

1. On the **Recording** toolbar, click **Insert Data Driven Commands** ().

   The test script recording pauses and the **Insert Data Driven Actions** page opens.

2. From the **Insert Data Driven Actions** page, use the mouse to drag the Object Finder () to the title bar, **Place an Order**, on the ClassicsCD application.

   Functional Test outlines the entire Place an Order dialog box with a red border.

3. Release the mouse button.

   The **Data Drive Actions** page opens. In the **Data Drive Actions** page, under the **DataDriven Commands** table, information appears about the objects you selected.

   You can place your mouse pointer over a row in this table to view the line of code that Functional Test inserts into the test script to data-drive the test script.

Now you are ready to begin Exercise 1.3: Adding Descriptive Headings to the Data.

Exercise 1.3: Adding Descriptive Headings to the Data

Before you begin, you must complete Exercise 1.2: Data-driving a Test.

In this exercise, you will add descriptive headings to the datapool you created in the previous lesson. Descriptive headings will make it easier to add data to the datapool.

1. In the **Data Drive Actions** page, in the **Data Driven Commands** table, in the first row, under the **Variable** header, click **ItemText** to select it.

   ItemText is the first descriptive heading under the **Variable** header. The first row appears selected.

2. Type **Composer** in the cell.
3. Click in the next cell beneath **Composer** and type **Item**.
4. Repeat step 3 to select each cell and type a descriptive name for each heading in the **Variable** field using the descriptive headings in the following table.

   Note: Do not use spaces in **Variable** names. Typically, you would look at the application to determine the appropriate headings for each row, but we have done that for you in the following table:

   Functional Test automatically updates the test script as you change each of the **Variable** names.

Variable
Composer
Item
Quantity
Card#
CardType
ExpDate
Name
Street
CityStateZip
Phone

5. Click **OK**.

   The **Insert Data Driven Actions** page closes.

Now the datapool has descriptive headings that make it easier to add more data. You will add more data to the datapool after you finish recording the test script.

Now you are ready to begin Exercise 1.4: Creating a Verification Point with a Datapool Reference to test that the total (amount due) for this order is correct.

Exercise 1.4: Creating a Verification Point with a Datapool Reference

Before you begin, you must complete Exercise 1.3: Adding Descriptive Headings to the Data.

In this exercise, you will create a verification point with a datapool reference to check that the total amount due for the order is correct in the Classics CD application.

A verification point captures object information and literal values from the application-under-test and stores it as the baseline for comparison during playback. When you play back the script, a verification point captures the object information again to compare it to the baseline and see if any changes have occurred, either intentionally or unintentionally. Comparing the actual object information in a script to the baseline is useful for identifying potential defects.

You will use a datapool reference instead of a literal value for the value that you are testing in the verification point. Using datapools with verification points gives you more flexibility to test realistic data with your test scripts.

Create a Verification Point with a Datapool Reference

1. On the **Recording** toolbar, click **Insert Verification Point or Action Command** ().

   The Verification Point and Action Wizard opens.

2. From the Verification Point and Action Wizard, use the mouse to drag the Object Finder () to **$19.99**, next to **Total**, in the Classics CD application.

   Functional Test outlines **$19.99** with a red border.

3. Release the mouse button.
4. On the **Select an Action** page, if necessary, click **Perform Data Verification Point** to test that the total amount equals the expected amount.
5. Click **Next**.
6. Click **Next**, again.
7. On the **Verification Point Data** page toolbar, click **Convert Value to Datapool Reference** () to use a datapool instead of a literal value in a verification point. (If you cannot see the **Convert Value to Datapool Reference** button on the toolbar, make the page larger by dragging a corner of the page.)

   The Datapool Reference Converter dialog box opens.

8. In the box, type **Total** to replace the **newvariable**
9. If necessary, select the **Add value to new record in datapool** check box to add the to the existing datapool record you created in the previous exercise.
10. Click **OK**.
11. Click **Finish**.

Place the Order and Close the ClassicsCD Application

1. In the ClassicsCD application click **Place Order** to place the order, and then click **OK** to close the message confirming your order.
2. Click **X** in the upper right corner of the Classics CD application to close the application.

Stop Recording

1. On the **Recording** toolbar, click **Stop Recording** () to write all recorded information to the test script.

The test script appears in the editor window.

Now you are ready to begin Exercise 1.5: Adding Data to the Datapool.

Exercise 1.5: Adding Data to the Datapool

Before you begin, you must complete Exercise 1.4: Creating a Verification Point with a Datapool.

In this exercise, you will add data to the datapool to test that the ClassicsCD sample application correctly totals each order placed in the application.

1. In the **Script Explorer**, double-click **Test Datapool** and then double-click **Private Test Datapool**. Under the test script editor, double-click the **Test Datapool** tab to expand the datapool editor so that you can work.

   The datapool editor opens and should look similar to the following table:

	Composer	Item	Quantity	Card#	Cardtype	ExpDate	Name	Street	CityStateZi
0	Schubert	String Quartets Nos. 4 & 14	1	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212

2. Position your mouse pointer in the datapool editor, right-click, and then click **Add Record**. Click **OK** to add a row after the first row.
3. For Functional Test, VB.NET Scripting, click in the datapool editor under row 0, and then right-click **Add Record**.
4. To add a second empty row, right-click **Add Record.**

   To save time, copy the data from row 0 in the datapool into the two empty rows that you created in step 2 and 3 to add data to the two empty rows.

5. Position the mouse pointer in the row 0 cell, right-click, and then click **Copy**.
6. Position the mouse pointer in row 1 cell, right-click, and then click **Paste**.
7. Click **Yes** to paste the data into the empty row.
8. Position the mouse pointer in row 2 cell, right-click, and then click **Paste**.
9. Click **Yes** to paste the data into the empty row.

   Change the value in the **Quantity** and **Total** columns to test that ClassicsCD sample application correctly totals each order:

10. In row 1, under the **Quantity** column, select the cell and type **2**.
11. In row 1, under the **Total** column, select the cell and type **$38.98**.
12. In row 2, under the **Quantity** column, select the cell and type **3**.
13. In row 2, under the **Total** column, select the cell and type **$57.97**.

    The data in the datapool should look like the following table:

	Composer	Item	Quantity	Card#	CardType	ExpDate	Name	Street	CityStateZi
0	Schubert	String Quartets Nos. 4 & 14	1	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212
1	Schubert	String Quartets Nos. 4 & 14	2	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212
2	Schubert	String Quartets Nos. 4 & 14	3	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212

14. On the **Test Datapool** tab, click **X** to close the datapool editor, and then click **Yes** to save the changes you made to the datapool.

Now you are ready to begin Exercise 1.6: Playing Back the Test.

Exercise 1.6: Playing Back the Test

Before you begin, you must complete Exercise 1.5: Adding Data to the Datapool.

In this exercise, you will play back the test you just recorded to see how easy it is to use a variety of data from a datapool to test the application.

Each time you play back a script with an associated datapool, the script accesses one record in the datapool. When you create a datapool reference for a verification point, the verification point uses the datapool reference to access a variable in that record. During playback, Functional Test substitutes the variable in the datapool for the datapool reference and compares the variable in the datapool to the actual results.

During playback you can view the script name, the script line number executing, status icons, and a description of the action in progress in the Playback Monitor.

1. To play back the test script, click **Script > Run**.

   The Select Log dialog box opens.

2. Click **Next**.
3. Click the **Datapool iterator Count** arrow and scroll up to select **Iterate Until Done** to access all three records in the datapool.
4. Click **Finish** to use the default log name.

   The Functional Test window minimizes, and the Playback Monitor appears in the top right area of your screen. Messages appear in the Playback Monitor as Functional Test plays back all of the recorded actions in the test script and enters data from the datapool.

   When the test script finishes playing back, Functional Test displays a log with the test results. A log is a file that contains the record of events that occur while playing back a script. A log includes the results of all verification points executed that can be used to test the application.

5. Click **X** to close the log.

Finish your tutorial by reviewing the materials in the Summary.

Creating Data-Driven Tests Summary

This tutorial has shown you how to create data-driven tests.

You have data-driven a test script, created descriptive headings for the data collected, added data to the datapool, created a data verification point with a datapool reference, played back a test script, and viewed the log.

Completed Learning Objectives

If you have completed all of the exercises, you should now be able to:

- Create a project and record a test script
- Data-drive a test
- Add descriptive headings to the data
- Create a verification point with a datapool reference
- Add data to the datapool
- Play back the test

More information

If you want to learn more about the topics covered in this tutorial, see the *Data-Driving Tests* section of the Functional Test Help.