



データ駆動型の機能テストの作成

目次

データ駆動型の機能テストの作成 1

概要: データ駆動型テストの作成 1

演習 1: プロジェクトの作成およびスクリプト・テストの記録. 2

 プロジェクトの作成 3

 記録の開始 3

 ClassicsCD アプリケーションの開始 3

演習 2: テストのデータ駆動 4

演習 3: データへの記述見出しの追加 4

演習 4: データ・プール参照を使用した検査ポイントの作成 5

 データ・プール参照を使用した検査ポイントの作成 5

 注文を出して ClassicsCD アプリケーションを閉じる 6

 記録の停止 6

演習 5: データ・プールへのデータの追加 6

演習 6: テストの再生 7

要約: データ駆動型テストの作成 8

データ駆動型の機能テストの作成

このチュートリアルでは、Rational® Functional Tester データ・ドライバー・ウィザードを使用して、データ駆動型の機能テストを作成する方法を学習します。

データ駆動型テストでは、データとテスト・スクリプトの間に 1 つの抽象層が設けられており、テスト・スクリプト中にはリテラル値がありません。データがテスト・スクリプトから分離されているので、以下を行うことができます。

- テスト・スクリプトに影響を与えずにテスト・データを変更する
- テスト・スクリプトではなくデータを変更することによって、新規のテスト・ケースを追加する
- 多数のテスト・スクリプトでテスト・データを共有する

学習目標

このチュートリアルを完了後、以下が可能になります。

- プロジェクトの作成およびテスト・スクリプトの記録
- テストのデータ駆動
- データへの記述見出しの追加
- データ・プール参照を使用した検査ポイントの作成
- データ・プールへのデータの追加
- テストの再生

必要な時間

30 分。

関連情報



PDF 版の表示

チュートリアル: 機能テストの作成

チュートリアル: キーワードに基づくマニュアル・テストの自動化

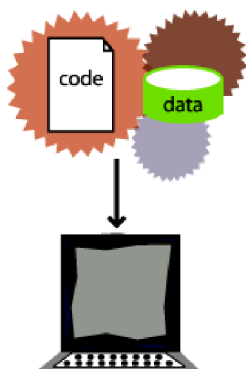
概要: データ駆動型テストの作成

このチュートリアルでは、Rational Functional Tester データ・ドライバー・ウィザードによって、アプリケーションをテストするためのさまざまな現実的データを使用してデータ駆動型のテストを作成する方法を学習します。

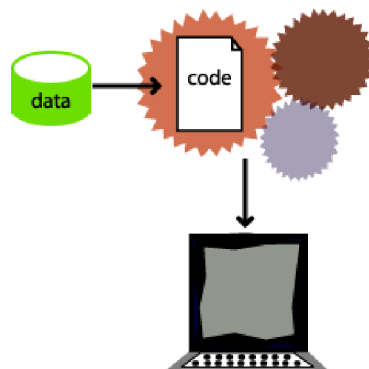
ClassicsCD サンプル・アプリケーションを使用して、プロジェクトの作成およびテスト・スクリプトの記録を行い、ClassicsCD サンプル・アプリケーションが注文の合計を正しく計算していることを検査します。さらに、データ・プール参照を使用した検査ポイントを作成して、Classics CD アプリケーションでの注文の合計金額が正しいことを検査します。

データ・プールについてさらに学習したい方に: データ・プールは、関連したデータ・レコードの集合体です。データ・プールは、テスト・スクリプトの再生中にテスト・スクリプト内の変数にデータ値を提供します。データ駆動型テストでは、テストへの入力として外部ファイルであるデータ・プールのデータを使用します。

左の図は、データをテスト・スクリプト内にハードコーディングされたりテラル参照と共に使用するテスト・スクリプトを示しています。右の図は、外部ファイルであるデータ・プールからのデータを使用する、データ駆動型のテスト・スクリプトを示しています。



ハードコーディングされたテスト・スクリプト



データ駆動型のテスト・スクリプト

学習目標

このチュートリアルを完了後、以下が可能になります。

- プロジェクトの作成およびテスト・スクリプトの記録
- テストのデータ駆動
- データへの記述見出しの追加
- データ・プール参照を使用した検査ポイントの作成
- データ・プールへのデータの追加
- テストの再生

注: 開始する前にチュートリアルを印刷して、演習を行う際にその印刷物を使用することを検討してください。チュートリアルの PDF 版を印刷するか、または各トピックの内部を右クリックしてから「印刷」をクリックすることにより個々の演習を印刷することができます。

必要な時間

このチュートリアルを完了するには約 30 分かかります。このチュートリアルに関連した他の概念も調べる場合には、完了するまでにさらに多くの時間がかかります。

演習 1: プロジェクトの作成およびスクリプト・テストの記録

この演習では、Classics CD サンプル・アプリケーションを使用して新規プロジェクトを作成し、テストの記録を開始することにより、購入された音楽 CD の合計金額をサンプル・アプリケーションが正しく計算することを検証します。

プロジェクトとは: プロジェクトとは、1 つ以上のソフトウェア・コンポーネントのテストを容易にするための、テスト・スクリプト、オブジェクト・マップ、検査ポイント、データ・プールなどのテスト資産の集合体です。テストを記録するには、その前に Functional Tester プロジェクトを作成しておく必要があります。


プロジェクトの作成

Classics CD サンプル・アプリケーションのテストに必要なテスト資産を保管するためのプロジェクトを作成します。

1. 「ウィンドウ」 → 「パースペクティブを開く」 → 「その他」をクリックして、機能テスト・パースペクティブを開きます。「パースペクティブを開く」ダイアログ・ボックスで、「**Functional Test**」オプションを選択します。
2. 「ファイル」 → 「新規」 → 「**Functional Test プロジェクト**」をクリックします。
3. 新規プロジェクトの名前として、DataDriveTutorial と入力します。
4. 「終了」をクリックします。

記録の開始

テスト・スクリプトの記録を開始して、顧客が音楽 CD を注文したときに、クレジットカードに課金される合計額がアプリケーションにリストされた正しい総額と同じであることを検証します。


1. 「Functional Test」ツールバーで、「**Functional Test スクリプトの記録**」()をクリックします。
2. テスト・スクリプトの名前として、OrderTotal と入力します。
3. 「次へ」をクリックします。

テスト・スクリプトを作成すると、Functional Tester はテスト・データ・プールおよび他のテスト資産を作成します。「専用テスト・データ・プール」および「順次」のデフォルトを使用してください。専用テスト・データ・プールは 1 つのスクリプトとだけ関連付けられていて、その他のスクリプトからは使用できません。順次の順序を使用すると、テスト・スクリプトはデータ・プール・レコードにデータ・プール内で出現する順序でアクセスします。

4. 「終了」をクリックします。「Functional Tester」ウィンドウが最小化され、記録モニターが開きます。



ClassicsCD アプリケーションの開始

ClassicsCD アプリケーションを開始して、アプリケーションをナビゲートし、データ駆動型で実行するダイアログ・ボックスを表示させます。

1. 「記録中」ツールバーで、「**アプリケーションの開始**」()をクリックします。
2. 必要であれば、「**アプリケーション名**」矢印をクリックしてオプションを表示してから、「**ClassicsJavaA - java**」を選択します。
3. 「OK」をクリックします。**ClassicsJavaA** は、Functional Tester に付属のサンプル・アプリケーション ClassicsCD のビルド 1 です。
4. ClassicsCD アプリケーションで、「**Composers**」の下にある「**Schubert**」をダブルクリックしてその作曲家の販売中 CD のリストを開いてから、「**String Quartets Nos. 4 & 14**」をクリックします。
5. 「**Place Order**」をクリックします。
6. 「OK」をクリックして、「Member Logon」ウィンドウを閉じます。
7. 「Place an Order」ウィンドウで、「**Card Number**」フィールドに 1234567890 と入力してから、「**Expiration Date**」フィールドに 09/09 と入力します。

演習 2: テストのデータ駆動

この演習では、データ・ドライバを使用してサンプル・アプリケーションからデータ・プールにデータを取り込みます。データ・プールは、関連したデータ・レコードの集合体です。データ・プールは、テスト・スクリプトの再生中にテスト・スクリプト内の変数にデータ値を提供します。

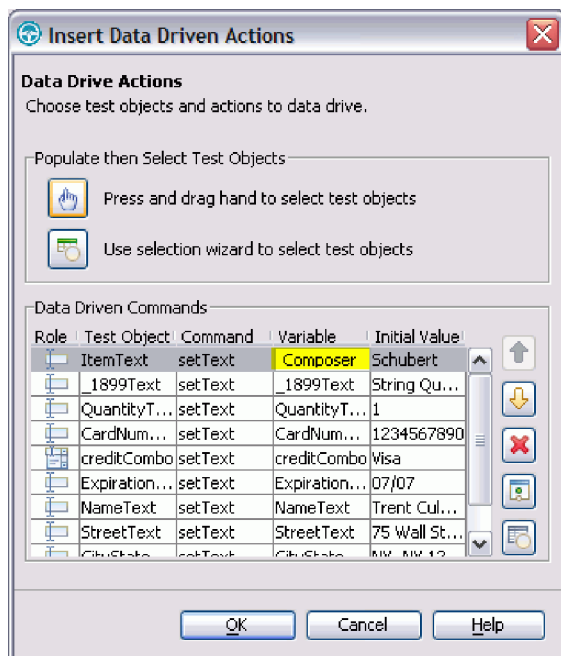
1. 「記録中」ツールバーで、「データ駆動型コマンドの挿入」() をクリックします。記録は一時停止します。
2. 「データ駆動型アクションを挿入します」ページで、マウスを使用してオブジェクト・ファインダー () を **ClassicsCD** アプリケーションの「**Place an Order**」ウィンドウのタイトル・バーにドラッグします。Functional Tester は、「Place an Order」ウィンドウ全体を赤色の枠で縁取ります。
3. マウス・ボタンを離します。「データ駆動型アクション」ページの「データ駆動型コマンド」テーブルの下に、選択したオブジェクトに関する情報が表示されます。

この表の行の上に移動すると、テスト・スクリプトをデータ駆動型で実行するために Functional Tester がテスト・スクリプトに挿入するコードの行を表示することができます。

演習 3: データへの記述見出しの追加

この演習では、直前の演習で作成したデータ・プールに記述見出しを追加します。記述見出しにより、データ・プールにデータを追加することが容易になります。

1. 「データ駆動型コマンド」テーブルの「変数」見出しで、**ItemText** を Composer に置き換えます。



2. これを順番に繰り返し、「変数」列の各セルを「変数」フィールドの各見出しの記述名に置き換えます。次の変数リストにあるテキストを記述名として使用してください。

注: 「変数」の名前内にはスペースを使用しないでください。通常はアプリケーションを観察して各行の適切な見出しを決定しますが、このチュートリアルではそれらをすでに次の変数リストのように決定してあります。

変数
Composer
Item
Quantity
CardNo
CardType
ExpiryDate
Name
Street
CityStateZip
Phone

それぞれの「変数」名を変更するとき、Functional Tester はテスト・スクリプトを自動的に更新します。

3. 「OK」をクリックします。

これで、データ・プールはデータの追加を容易にする記述見出しを持つようになりました。データ・プールへのデータの追加は、テスト・スクリプトの記録を終了した後に行います。



演習 4: データ・プール参照を使用した検査ポイントの作成


この演習では、データ・プール参照を使用した検査ポイントを作成して、Classics CD アプリケーションでの注文の合計金額が正しいことを検査します。

検査ポイントとは: 検査ポイントは、オブジェクト情報およびリテラル値をテスト対象のアプリケーションからキャプチャーして、それを再生時の比較のベースラインとして保管します。そのスクリプトを再生するとき、検査ポイントは再びオブジェクト情報をキャプチャーしてベースラインと比較し、意図的な変更や意図しない変更が加えられたかどうかを調べます。スクリプト内の実際のオブジェクト情報をベースラインと比較することは、潜在的な欠陥を識別するために役立ちます。

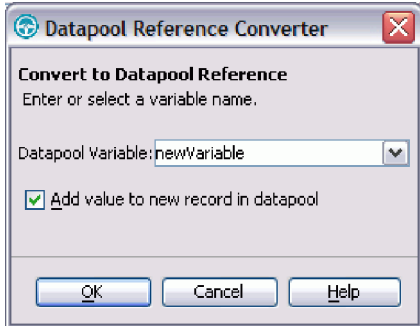
検査ポイントでテストする値には、リテラル値ではなくデータ・プール参照を使用します。データ・プールを検査ポイントで使用すると、テスト・スクリプトで現実的なデータをより柔軟にテストできるようになります。

データ・プール参照を使用した検査ポイントの作成

1. 「記録中」ツールバーで、「検査ポイント・コマンドまたはアクション・コマンドの挿入」() をクリックします。
2. 検査ポイントおよびアクション・ウィザードで、マウスを使用して「オブジェクト・ファインダー」() を Classics CD アプリケーションの「Total」の隣にある「\$19.99」にドラッグします。Functional Tester は、赤色の枠で「\$19.99」を縁取ります。
3. 「アクションの選択」ページが表示されない場合、「次へ」をクリックします。
4. 「アクションの選択」ページで、「データ検査ポイントの実行」をクリックして合計金額が预期される金額と等しいかどうかをテストします。
5. 「次へ」をクリックします。
6. 「検査ポイント・データ・コマンドの挿入」ページで、「次へ」をクリックします。

7. 「検査ポイント・データ」ページのツールバーで、「データ・プール参照への値の変換」 () をクリックして、検査ポイントでリテラル値ではなくデータ・プールを使用するようにします。(ツールバー上に「データ・プール参照への値の変換」ボタンが見えない場合、ページの角をドラッグしてページを拡大してください)。

「データ・プール参照コンバーター」ダイアログ・ボックスが開きます。




8. 「データ・プール変数」フィールドで、データ・プールの見出しとして Total と入力して「newVariable」を置き換えます。
9. 「データ・プール内の新規レコードに値を追加」チェック・ボックスを選択して「Total」を直前の演習で作成した既存のデータ・プール・レコードに追加します。
10. 「OK」をクリックします。
11. 「終了」をクリックします。

注文を出して ClassicsCD アプリケーションを閉じる

1. ClassicsCD アプリケーションで、「Place Order」をクリックして注文を出してから、「OK」をクリックして注文を確認するメッセージを閉じます。
2. Classics CD アプリケーションの右上の角にある「X」をクリックして、アプリケーションを閉じます。

記録の停止

「記録中」ツールバーで、「記録の停止」 () をクリックして、すべての記録済み情報をテスト・スクリプトに書き込みます。

テスト・スクリプトがエディター・ウィンドウに表示されます。

演習 5: データ・プールへのデータの追加

この演習では、データをデータ・プールに追加して、ClassicsCD サンプル・アプリケーションがアプリケーション内で出された各注文の合計を正しく計算しているかどうかをテストします。

1. スクリプト・エクスプローラーで、「テスト・データ・プール」をダブルクリックしてから、「専用テスト・データ・プール」をダブルクリックします。テスト・スクリプト・エディターで、「テスト・データ・プール」タブをダブルクリックし、データ・プール・エディターを展開して作業できるようにします。

データ・プール・エディターが開きます。以下の表のように表示されるはずです。

	Composer	Item	Quantity	Card#	CardType	ExpDate	Name	Street	CityStZip	Phone	Total
0	Schubert	String Quartets Nos. 4 & 14	1	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212	212-552-1867	\$19.99

- マウス・ポインターをデータ・プール・エディター上に置いて、右クリックしてから、「**レコードの追加**」をクリックします。「**OK**」をクリックして、最初の行の後に行を追加します。
- 2 番目の空の行を追加するために、右クリックして「**レコードの追加**」を選択します。時間を節約するために、データ・プール内の行 0 から、ステップ 2 および 3 で作成した 2 つの空の行にデータをコピーします。
- マウス・ポインターを行 0 のセル上に置いて、右クリックしてから、「**コピー**」をクリックします。
- マウス・ポインターを行 1 のセル上に置いて、右クリックしてから、「**貼り付け**」をクリックします。
- 「**はい**」をクリックして、データを空の行に貼り付けます。
- マウス・ポインターを行 2 のセル上に置いて、右クリックしてから、「**貼り付け**」をクリックします。
- 「**はい**」をクリックして、データを空の行に貼り付けます。
- 次のように「**Quantity**」列および「**Total**」列の値を変更して、ClassicsCD サンプル・アプリケーションが各注文の合計を正しく計算するかどうかをテストします。
 - 行 1 の「**Quantity**」列で、セルを選択して 2 と入力します。
 - 行 1 の「**Total**」列で、セルを選択して \$38.98 と入力します。
 - 行 2 の「**Quantity**」列で、セルを選択して 3 と入力します。
 - 行 2 の「**Total**」列で、セルを選択して \$57.97 と入力します。

データ・プール内のデータは、次の表のようになるはずです。

	Composer	Item	Quantity	Card#	CardType	ExpDate	Name	Street	CityStZip	Phone	Total
0	Schubert	String Quartets Nos. 4 & 14	1	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212	212-552-1867	\$19.99
1	Schubert	String Quartets Nos. 4 & 14	2	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212	212-552-1867	\$38.98
2	Schubert	String Quartets Nos. 4 & 14	3	1234567890	Visa	09/09	Trent Culpito	75 Wall St.	Ny, Ny 12212	212-552-1867	\$57.97

- 「**テスト・データ・プール**」タブで、「**X**」をクリックしてデータ・プール・エディターを閉じてから、「**はい**」をクリックしてデータ・プールに加えた変更を保管します。

演習 6: テストの再生

この演習では、先程記録したテストを再生して、データ・プールからのさまざまなデータを使用してアプリケーションをテストすることが容易であることを確認します。

スクリプトに関連するデータ・プールと共に再生するたびに、スクリプトはデータ・プール内の 1 つのレコードにアクセスします。検査ポイントのデータ・プール参照を作成するとき、検査ポイントはデータ・プール参照を使用してそのレコード内の変数にアクセスします。再生の際に、Functional Tester はデータ・プール参照をデータ・プール内の変数に置き換えて、データ・プール内の変数を実際の結果と比較します。

再生の際に、スクリプト名、実行中のスクリプト行番号、状況アイコン、および進行中のアクションに関する記述が、再生モニター内に表示されます。

1. テスト・スクリプトを再生するには、「スクリプト」 → 「実行」をクリックします。
2. 「ログの選択」ウィンドウで、「次へ」をクリックします。
3. 「データ・プール反復カウント」矢印をクリックしてから、スクロールして「完了するまで繰り返します」を選択し、データ・プール内の 3 つのレコードすべてにアクセスします。
4. デフォルトのログ名を使用するために、「終了」をクリックします。

「Functional Tester」ウィンドウが最小化し、画面の右上の領域に再生モニターが表示されます。

Functional Tester がテスト・スクリプト内に記録されたアクションをすべてを再生して、データ・プールからのデータを入力すると、再生モニターにメッセージが表示されます。

テスト・スクリプトの再生が完了すると、Functional Tester はテスト結果を含むログを表示します。ログは、スクリプトの再生中に生じたイベントの記録を含むファイルです。ログには、アプリケーションのテストに使用できる、実行されたすべての検査ポイントの結果が含まれます。

5. 「X」をクリックしてログを閉じます。

要約: データ駆動型テストの作成

このチュートリアルでは、データ駆動型テストの作成方法を示しました。

データ駆動型テストのスクリプトを作成し、収集したデータの記述見出しを作成し、データ・プールにデータを追加し、データ・プール参照を使用したデータ検査ポイントを作成し、テスト・スクリプトを再生し、ログを表示しました。

学習した演習

このチュートリアルを完了すると、以下を行う方法を学習したことになります。

- プロジェクトの作成およびテスト・スクリプトの記録
- テストのデータ駆動
- データへの記述見出しの追加
- データ・プール参照を使用した検査ポイントの作成
- データ・プールへのデータの追加
- テストの再生

追加リソース

このチュートリアルで扱われたトピックについてもっと詳しく知りたい場合、Functional Tester のヘルプでデータ駆動テストのセクションを参照してください。

関連情報

 [ibm.com](https://www.ibm.com)

 [eclipse.org](https://www.eclipse.org)