***C. Tìm Hiểu Công Cụ***

***IBM Rational Functional Tester***

**I. Giới thiệu công cụ:**

IBM Rational Functional Tester là phần mềm kiểm thử chức năng, một dụng cụ thử nghiệm hồi quy tiên tiến, được tự động hóa cho tester và người phát triển GUI(Graphical User Interface). Là những người cần kiểm soát cấp cao hơn cho việc test với công nghệ java, Microsoft® Visual Studio .NET, và ứng dụng Web-based. Chạy trên nền win và linux.

Rational Funtional Tester ghi lại sự tương tác trong lịch trình của những người làm việc với Java, Web, Visual Studio.Net, trên trình ứng dụng Win Form, và Web- Form tạo ra cho việc kiểm thử một kịch bản, bằng cách mô phỏng trở lại những thao tác đã được thực hiện.

Trong lúc đó hình ảnh sẽ được ghi lại, người sử dụng có thể lồng vào thời gian xác định trong những trích đoạn theo lý thuyết mà dữ liệu đưa ra hoặc những đặc tính mà trình ứng dụng chưa đạt đến sẽ kiểm tra được trong quá trình kiểm thử. Trong quá trình quay lại, có những thời điểm xác minh các vấn đề đã thực hiện và sẽ so sánh với những thông tin được ghi chép đảm bảo theo đúng những thông tin được ghi chép. Sau đó việc kiểm thử sẽ được ghi hình một cách linh hoạt, những người kiểm thử có thể xác định được sự lựa chọn ngôn ngữ để viết cho khách hàng dựa vào kịch bản, tới những việc đã vượt quá nhiệm vụ cần thực hiện, bao gồm những dữ liệu thao tác bằng tay và những yêu cầu về cấu hình máy tính, những vấn đề này bảo đảm cho việc kiểm thử được thực hiện đúng đắn và có thể vận hành được sự kiểm thử. Sau khi thực hiện xong quá trình kiểm thử.

Rational Funtional Tester sẽ phát sinh ra một bảng báo cáo về những kết quả đạt được trong quá trình kiểm thử và nó dùng để so sánh với những thời điểm xác định.

Với việc sử dụng Rational Funtional Tester đội dự án có nhiều điều chắc chắn về những vấn đề nó được bộc lộ một cách hiệu quả trong nhiều trình ứng dụng phức tạp, làm tăng dần cơ hội cho việc bắt được những khuyết điểm và được phục hồi trước khi những sản phẩm được đưa ra.

**II. Lợi ích sử dụng công cụ:**

* Tạo sự tin cậy cho chúng ta trong việc kiểm thử các phần mềm dùng các ngôn ngữ như Java, Wed, Visual Studio.Net trên trình ứng dụng Win-Form và Web-Form.
* Là sự lựa chọn cho các phần mềm dùng ngôn ngữ Java hoặc Visual Basic.Net, nó giúp tạo ra được những kịch bản kiểm thử để so sánh với bảng phân tích, xem có đúng theo yêu cầu của khách hàng không.
* Nơi Java, Visual Basic.Net được biên tập và gỡ rối bởi những người kiểm thử tiên tiến.
* Những kịch bản bằng công nghệ cung cấp thường xuyên sự thay đổi của giao diện người dùng.
* Sự tương quan của dữ liệu một cách tự động và những dữ liệu kiểm thử cần loại trừ sẽ tốt hơn nhiều so với việc kiểm thử bằng tay.
* Những mốc kiểm tra phù hợp với những khuôn mẫu đã đưa ra.
* Sự tiên tiến của năng lực được duy trì trong sơ đồ các đối tượng.
* Là nơi tin cậy của việc kiểm thử và thi hành trên nền Linux..

**Về những tính năng chính:**(About this release)

Những phần chính của IBM Rational Funtional Tester bao gồm những chức năng sau:

* Mở rộng chức năng kiểm thử cho những trình điều khiển SAP UI- Cho phép bạn kiểm thử những chức năng bên ngoài của những trình điều khiển được xây dựng trên nền SAP frameword.
* Mở rộng chức năng kiểm thử hổ trợ cho điều khiển Siebel 7.8
* Quy trình với Rational Process Advisor (RPA)-Giúp cho những người chưa biết và đang sử dụng những tiến trình quanh những thao tác đặt trưng sẽ thi hành được.
* Hổ trợ cho kiểm thử những ứng dụng trong Mozilla Firefox 1.5
* Sự tích hợp với ClearQuest Test Manager V7.0.0.1- Cho phép bạn chạy lại kịch bảng kiểm thử chức năng từ ClearQuest Test Manager, theo một nhịp độ tự động, và kiểm tra được những lỗi. Với sự tích hợp ClearQuest Test Manager, bạn có thể tích hợp kịch bảng kiểm thử chức năng Test Case và Cofigured Test Case và thi hành chúng. ClearQuest Test Manager cho phép bạn thi hành việc phân tích những kết quả và ghi lại những kết quả cho việc báo cáo và phân tích.
* IBM Interllation Manager – cung cấp nhanh và đơn giản những phương thức cài đặt, sửa đổi, cập nhật, và tháo bỏ gói cài đặt .
* Hổ trợ cho .Net Frameword 2.0 bao gồm hổ trợ cho những bản ghi và kiểm thử những trình ứng dụng chứa DataGridWiew và MaskeDTestBox.
* Hổ trợ cho TPTP- cho phép làm theo một nhịp độ tự động sử dụng TPTP.
* Hổ trợ cho việc điều khiển Eclipse 3.0.
* Hổ trợ cho việc kiểm thử trên môi trường Citrix.
* Hổ trợ cho việc xác minh những điểm tạm thời.

Những chức năng chính này sẽ được trình bày kỷ trong phần demo của chương trình.

**III. Điều kiện sử dụng công cụ:**

**Thông tin về cài đặt**

Thông tin về việc cài đặt IBM Rational Funtional Tester hổ trợ hệ thống hệ điều hành và đường dẫn, bạn sẽ nhìn thấy được trong khi cài đặt IBM Rational Funtional Tester.

**Yêu cầu về phần cứng**

1. Bộ xử lí yêu cầu mức tối thiểu Intel Pentium 4 1.4GHz.
2. Ram đề nghị và tối thiểu là 1GH.
3. Dung lượng ổ đĩa trống là 750MB.

**Yêu cầu về phần mềm kèm theo**

1. .Net FrameWord 1.0, 1.1, hoặc 1.2
2. Môi trường phát triển kết hợp tuỳ chọn(IDE)
   * VS.Net 2003 với .Net FrameWord 1.0 hay 1.1.

VS.Net 2005 với .Net FrameWord 1.2

**Hệ thống hệ điều hành**

Những hệ điều hành bên dưới hổ trợ cho việc cài đặt:

* Microsoft WindowXP Service Packs 1 và 2.
* Microsoft Window 2000 với Service Pack 4.
* Microsoft Window 2000 Server với Service Pack 4.
* Microsoft Window Server 2003 Standard Edition với Service Pack 1.
* Microsoft Window Server 2003 Eterprise Edition với Service Pack 1.
* Red Hat Eterprise Linux Wordstation Vertion 4.0.

**IV. Hướng dẫn sử dụng phần mềm:**

**1 Chức năng của những biểu tượng trên thanh Toolbar:**

Trên thanh toolbar có những command như sau:

**New**: Hiển thị ra một cái hộp để bạn tạo một Project mới bao gồm tạo Funtional Tester Project, một Record a Funtional Tester Scrip, Test Object map, hoặc Help Superclass. Bạn click vào những biểu tượng đó để thực những cách tạo mà bạn muốn hoặc bạn click vào biểu tượng và sau đó chọn phần bạn muốn tạo.



**Save** : Dùng để lưu những phần bạn đã làm.



**Print**: phần hiển thị của hộp print này giúp bạn in nhưng kịchbản hiện thời bạn đanglàm.



**Create a Functional Tester Project**: tại đây bạn có thể tạo cho mình một Project mới.



**Connect to an existing Functional Tester Project** :phần kết nối đến một Project khác trước khibạn sử dụng Project hiện thời.



**Create an Empty Functional Test Script**: Tạo một khịc bản trống và bạn có thể thêm vào những phần code của Java.



**Create a Test Object Map**: tạo một đối tượng kiểm thử dạng biểu đồ cho dự án của bạn,nó giống như một sơ đồ cây.



**Create a Test Datapool**: tạomotj hộp dử liệu kiểm thử



**Create a Test Folder**: tạo một danh mục kiểm thử.



**Record a Functional Tester Script**: Giúp bạn thấyđược những thông tin của kịch bản mới và chạy những mẫu tin.



**Insert Recording into Active Functional Tester Script**: chèn bào một bản ghi trong kịch bản đang hoạt động.



**Run Functional Tester Script** : dùng để chạy kịch bản, khi bạn đã chọn chương trình và chọn nút này thì RFT sẽ ghi lại tất cả những hoạt động mà bạn thực hiện, nếu sau khi đã hoàn thành xong việc chạy và bạnchọn tiêp nút này thì nó sẽ hiện ra một File như vidio quay lại tất cả những thao tác mà bạn đã thực hiện



**Debug Functional Tester Script**: Chạy một kịch bản hiện hành, chỉ ra lỗi và hiển thị những thông tin giúp bạn gỡ những lỗi mắc phải.



**Configure Applications for Testing** : Định cấu hình cho việc kiểm thử,tạo đây bạn có thể thêm và hiệu chỉnh những thông tin cấu hình như tên, đường dẫn, hoặc những thông tin khác trong Funtional Tester trong khi bạn chạy chương trình



**Enable Environments for Testing**: lựa chọn môi truờng cho việc kiểm thử.



**Open Test Object Inspector**: Mở một đối tượng kiểm thử nó sẽ xuất hiện dưới dạng hình cây.



**Insert Verification Point** **into Active Functional Tester Script:**



**Insert Test Object into Active Functional Tester Script** :



**Insert Data Driven Commands into Active Functional Tester Script**.



**Find Literals and Replace with Datapool Reference**



**External Tools**: xây dựng tool bên ngoài cho phép bạn chạy và định c cấu hình cho những tool bên ngoài.



**Search**: Với chức năng tìm kiếm bạn có thể tìm kiếm những vấn đ mà bạn cần.



**Last Edit Location**: quay lại vị trí cuối.



**Back**:quay lại phần trước.



**Forward**: tiếp đến phần tiép theo.



**2. Tổng quan Funtional Tester Project:**

Phần này nằm ở Pane bên trái của cửa sổ khi bạn tạo một Project, danh sách kiểm thử này nằm trong 1 project.

Dưới đây là một số biểu tượng xuất hiện trong pane của một Project.

Folders



Test Scripts



Shared test object maps



Log folders



Logs



Java™ file



Trên phần banner của một project xuất hiện những biểu tượng sau.

Kết nối đến một Project mới.



Refresh Projects .



Nút xảy ra đồng thời của sự biên tập.

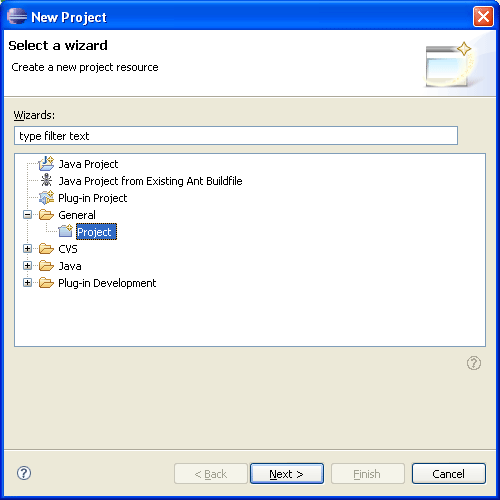


1. **Cách tạo một Project đơn giản:**

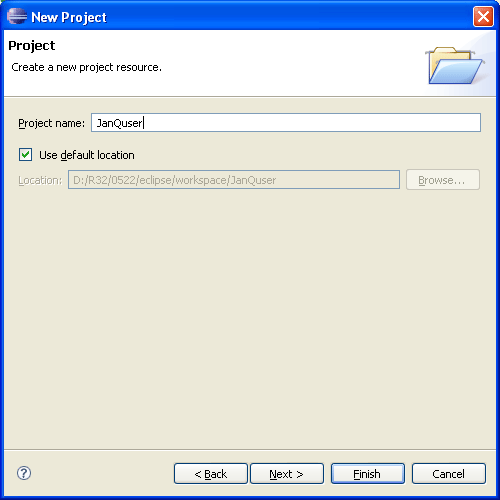
**Sử dụng File menu.**

Bạn có thể tạo ra một tài nguyên mới bằng cách vào **File->new.** Sau đo sẽ xuất hiện menu bar, bạn tạo một project đơn giản bằng cách như sau.

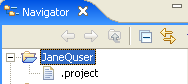
1. Từ menu bar bạn chọn **File->New->Project…**



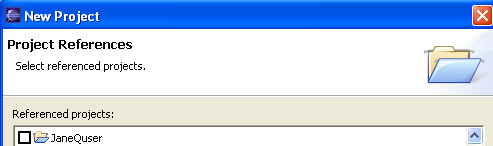
1. Cửa sổ như trên se xuất hiện,bạnchọn tab **General->Project->Next**
2. Trong trường **Project name**, bạn đặt tên cho project ví dụ như JeaQuser.
3. Nếu bạn không muốn thay đổi đường dẫn thì bạn check vào ô User default location.



Sau khi chọn Finish bạn sẽ nhìn thấy Project mà bạn đã tạo



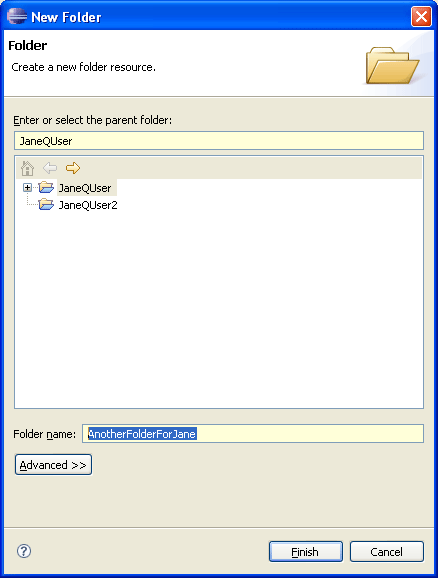
Nếu bạn muốn tạo một dự án thứ hai đặt tên là JaneQuser2 thì bạn vẫn làm như trên , nhưng thay vì bạn click nút Finish thì bạn click nút Next. Từ đó chúng ta muốn tạo 2 Project độc lập chúng ta sẽ chọn bất cứ nơi đâu của dự án trong cửa sổ **Project References.** Click **Finish** để hoàn thành việc tạo một Project đơn giản.



**Sử dụng Popup**

Bây giờ chúng ta đã có một Project,chúng ta sẽ tạo một Forder bằng cách sử dụng popup menu.

1. Từ Project JaneQuser bạn chon **New->Folder**
2. Trong New Folder Wizad, tên Project của bạn đã xuất hiện theo sự mặc định trong trường **Enter or select the parent folder** vì ngay từ đầu chúng ta đã chọn tạo Folder trong mục JaneQuser.
3. Trong **Folder name** bạn đặt tên tuỳ ý mà bạn muốn cho Folder của bạn.

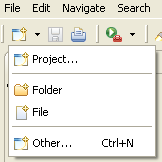


1. Click **Finish** để hoàn thành

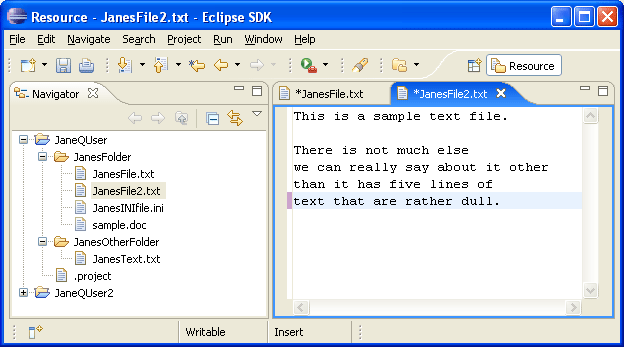
**Sử dụng một button mới:**

Bây giờ chúng ta sẽ tạo 1 File cho Project sử dụng cách thứ 3 đó là botton **new** trên thanh Toobar

1. Chọn 1 Folder tuỳ ý mà ta đã tạo ở trên.
2. Trên thanh Toolbar ta chọn ta chọn biểu tượng và chọn **File**



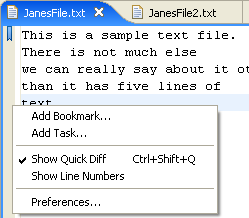
1. Trong trường File Name bạn đặttên tuỳ ý.
2. Click để hoàn thành.



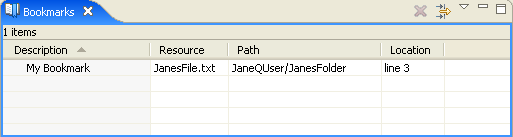
**Adding and Viewing Bookmarks**

Muốn thêm một Bookmark bạn làm như sau:

1. Từ menu bar bạn chon. **Window->Show view->Bookmarks**
2. Bạn chọn File đã có chẳng hạn như Janefile.txt
3. Sau đó bạn chọn **Add Bookmark**



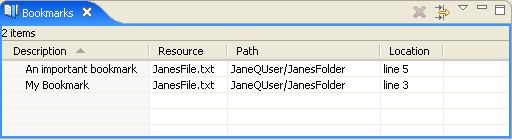
Một bookmark mới sẽ xuất hiện



1. Trong bảng này bạn chọn File có tên là JanesFile.txt. Từ thanh menu bạn chọn **Edit->Bookmarks.**

Chúng ta sẽ biết được những tên File sử dụng nó sẽ được mô tả cụ thể trên Bookmarks.

Bây giờ chúng ta quan sát sẽ thấy xuất hiện 2 hộp Bookmarks.



**4. Create a data-driven test(Tạo kiểm thử Data-Driven)**

Trong phần hướng dẫn này, bạn sẽ học cách tạo kiểm thử Data-Driven và cách sử dụng từ những sự đa dạng của dữ liệu kiểm thử trên trình ứng dụng với Rational Funtional Tester Data-Driven Wizard.

Sau khi hoàn thành xong phần này bạn sẽ làm được những việc như sau.

* Tạo một Project và cách ghi lại kịch bản kiểm thử.
* Kiểm thử Data-drive.
* Thêm một tựa đề mô tả vào dữ liệu.
* Tạo một điểm kiểm chứng với tham chiếu Datapool.
* Thêm dữ liệu trong Datapool.
* Chạy lại kịch bản kiểm thử.

1. **Tạo một Project**

Muốn tạo một Project bạn vào **Windows** > **Open Perspective** > **Other.** Sau đó bạn chọn **Functional Test.**

Click **File** > **New** > **Functional Test Project**.

Bạn đặt tên cho Project và sau đó chọn **Finish** để hoàn thành quá trình tạo một Project.

* **Bắt đầu cho một báo cáo.**

Trên thanh Toolbar của phần mềm Rational Test bạn chọn **Record a Functional Test Script**(). bạn đặt tên cho Scrip sau cho click **Next** và chọn phần bạn muốn sử dụng và click **Finish** để hoàn thành quá trình.



Cửa sổ nhỏ nhất của Window và màn hình Recording sẽ đư

* **Bắt đầu với trình ứng dụng ClassicsCD**

Trên thanh Toolbar của bảng Recording chọn **Start Application**(). Bạn click **Application Name** cho phép bạn có nhiều lựa chọn, và sau đó bạn chọn **ClassicsJavaA - java**.



Sau đó chọn **Ok**

Trong trình ứng dụng ClassicsCD ở bên dưới **Composers,** bạn click 2 lần vào **Schubert** . Sau đó bạn click **String Quartets Nos. 4 & 14**

Click **Place Order** và sau đó chọn **Ok** cửa sổ **Place an Order** xuất hiện.

Trong trình ứng dụng ClassicsCD, bạn gõ 1234567890 trong trường **Card Number** và gõ 09/09 trong trường **Expiration Date**

1. **Data-driving a test**

Trong phần này, bạn sẽ sử dụng Data-driver cư trú trên Datapool với dữ liệu có sẳn từ ví dụ trên trình ứng dụng. Một Datapool là một tập hợp những dữ liệu báo cáo liên quan. Một Datapool sẽ cung cấp giá trị của dữ liệu và bạn dùng nó để so sánh với sự biến đổi trong kịch bản kiểm thử trong lúc chạy lại kịch bản kiểm thử.

* Trên thanh Toolbar của cửa sổ Recording, click **Insert Data Driven Commands** (). Việc ghi lại sẽ tạm ngưng và bản **Insert Data Driven Commands** được mở.



* Từ trang **Insert Data Driven Commands,** sử dụng chuột di chuyển đến Object Finder () tới thanh Title bar của cửa sổ **Place an Order** trên trình ứng dụng **ClassicsCD.**



Funtional Tester sẽ xuất ra toàn bộ bảng phát thảo của cửa sổ Place an Order với đường viền màu đỏ.

* Sự thoát khỏi nút chuột(Release the mouse button)

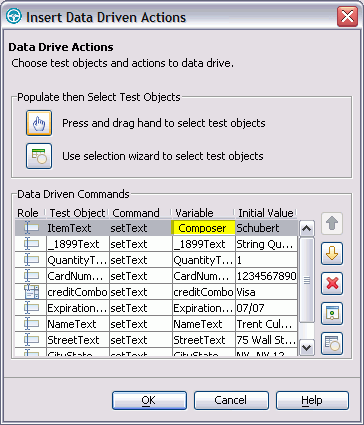
Trang **The Data Driver Actions** được mở, trong trang **The Data Driver Actions,** dưới bảng nút lệnh **data driver** thông tin về những mục bạn chọn sẽ được xuất hiện.

Bạn nhìn qua những dòng trong bảng hiển thị bạn sẽ thấy đó là những dòng code mà Rational Tester dùng để chèn vào kịch bản kiểm thử tới Data-Driver kịch bản kiểm thử.

**3. Adding descriptive headings to the data(Thêm vào mô tả tựa đề tới dữ liệu)**

Trong phần này, bạn sẽ thêm vào mô tả tựa đề mà bạn đã làm ở phần trước. Mô tả tựa đề giúp bạn dễ dàng hơn trong việc thêm dữ liệu ở Datapool

Trong bảng **Data Driven Commands**, sau khi xuất hiện bảng tiêu đề **Variable**, nó sẽ thay thế **ItemText** với người biên soạn.



Sự lặp lại liên tục, thay thế cho mỗi ô trong bảng **Variable** với việc thay thế tên cho mỗi tiêu đề trong trường **Variable.** Sử dụng những đề mục trong danh sách bên dưới để thực hiện việc thay thế tên.  
 Click **Ok,** trang **Insert Data Driver Action** sẽ được đóng lại.

Bây giờ, Datapool đã mô tả tựa đề nó giúp bạn dễ dàng hơn trong việc thêm dữ liệu. Bạn sẽ thêm nhiều dữ liệu trong Datapool sau khi hoàn thành việc ghi lại của kịch bản kiểm thử.

**4. Creating a verification point with a datapool reference(Tạo một điểm kiểm chứng với tham chiếu Datapool)**

Trong phần này, chúng ta sẽ tạo một điểm kiểm chứng với tham chiếu Datapool để kiểm tra tất cả những vấn đề xem nó có đúng hay không trên trình ứng dụng ClassicsCD.

Chúng ta cần tìm hiểu điểm kiểm chứng(**verification point** ) là gì?

*Điểm kiểm chứng:* Đó là sự giữ lại thông tin của những mục tiêu và những giá trị bằng chữ từ trình ứng dụng sau khi kiểm thử và chúng ta cần lưu giữ nó giống như một cái cơ sỡ để dùng trong việc so sánh trong lúc chạy lại kịch bản. Khi bạn chạy lại kịch bản, điểm kiểm chứng sẽ ghi lại thông tin của những mục tiêu trở lại sau đó chúng ta dùng nó để so sánh với những điểm cơ sỡ, và sẽ nhìn thấy được những vấn đế đúng sai so với kịch bản.

**Tạo một điểm kiểm chứng với tham chiếu Datapool.**

1. Trên thanh Toolbar của bảng **Recording** ,click **Insert Verification Point or Action Command** (). The Verification Point and Action Wizard sẽ được mở.



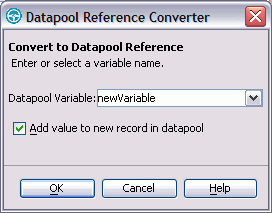
1. Từ The Verification Point and Action Wizard, rê chuột đến **Object Finder** () to $19.99, bạn chọn "Total" trong trình ứng dụng ClassicsCD.



1. Funtional Tester $19.99 sẽ xuất hiện với dòng màu đỏ.
2. Sau đó bạn click **Next.**
3. Và lại tiếp tục Click **Next.**
4. Trên thanh Toolbar của trang Verification Point Data, Click **Convert Value to Datapool Reference** ().



Sau đó sẽ hiển thị ra hộp thoại sau:



1. Trong tường Datapool Variable bạn chọn **newVariable** để thay đổi toàn bộ tiêu đề trong Datapool.
2. Nếu cần thiết bạn có thể chọn **Add value to new record in datapool để** thêm giá trị tới Record mới trong Datapool.
3. Click **Ok.**
4. Click **Finish.**

Place the order and close the ClassicsCD application(Sắp đặt thứ tự và đóng trình ứng dụng ClassicsCD)

1. Trong trình ứng dụng **ClassicsCD** Click Place the order, và sau đó click **Ok**  để đóng hộp thoại chứng thực.
2. Click **X** ở góc bên phải của trình ứng dụng **ClassicsCD** để đóng trình ứng dụng.

Stop recording

Trên thanh Toolbar của bảng **Recording,** click **Stop Recording** (), để viết tất cả các thông tin trên bảng ghi tới kịch bản kiểm thử.



Kịch bản kiểm thử sẽ hiển thị trên bảng trên trình sạon thảo Window.

**5. Adding data to the datapool(Thêm dữ liệu cho Datapool)**

1. Trong Script Explorer, double-click **Test Datapool** và sau đó double-click **Private Test Datapool,** trong trình soạn thảo kịch bản kiểm thử, double-click the **Test Datapool** trên trình soạn

thảo mở rộng mà bạn muốn làm việc.

2. Click phải và sau đó click **Add Record** , Click **Ok** để thêm vào hàng trống đầu tiên.

1. Tạo thêm vào hàng trống thứ hai, bạn lại Click **Add Record**

Lưu lại,Copy dữ liệu từ hàng 0 trong Datapool đến hàng trống thứ 2 mà bạn đã làm ở bước 2 và 3.

1. Tại vị trí hàng 0, bạn click chuột phải, chọn **Copy.**
2. Tại vị trí hàng 1 , bạn click chuột phải và chọn **Paste.**
3. Click **Yes** để dán dữ liệu vào hàng trống.
4. Tương tự như vậy ở hàng 2.
5. Trong **Test Datapool** click **X** phía bên phải cửa sổ, Click **Yes** để lưu những chuyển đổi mà bạn đã làm trong Datapool.

**6**. **Playing back the test(Chạy lại quá trình kiểm thử)**

Trong phần này, chúng ta sẽ biết cách chạy lại kịch bản kiểm thử, bạn sẽ nhìn thấy tất cả những điều mà bạn đã làm một cách dễ dàng với việc sử dụng dữ liệu từ Datapool để kiểm thử trên trình ứng dụng.

1. Để chạy lại kịch bản, click **Script** > **Run.** Cửa sổ Select Log sẽ được mở.
2. Click **Next.**
3. Click mũi tên **Datapool Iteration Count** và sau đó chọn **Iterate Until Done** để truy xuất tất cả 3 bảng ttrong Datapool.
4. Click **Finish** để sử dụng tên hiển thị mặc định.

Cửa sổ Funtion Tester hiện ra, và màn hình PlayBack sẽ hiển thị trên màn hình chính của bạn. Một hộp thoại xuất hiện trong màn hình PlayBack giống như việc chạy lại tất cả các bản ghi trong kịch bản kiểm thử .

Khi kịch bản kiểm thử chạy lại hoàn thành, Funtion Tester sẽ hiển thị một bảng ghi với những kết quả kiểm thử. Bảng ghi này là một File ghi lại tất cả các sự kiện xảy ra khi bạn chạy kịch bản. Bảng ghi bao gồm những kết quả của tất cả những điểm kiểm chứng mà bạn đã thực hiện trên trình ứng dụng.

1. Click **X** để đóng cửa sổ Log.

**7. Cách tạo một Funtional Test**

Rational Funtional Tester sẽ hướng dẫn bạn một cách tổng quan trong việc tạo và chạy lại kiểm thử chức năng. Đây là một sự hướng dẫn toàn diện sử dụng những ví dụ được cài đặt trong trình ứng dụng của Java với những sản phẩm hoàn thiện.

**Mục tiêu của phần này.**

Sau khi hoàn thành những phần này, bạn sẽ làm được những vấn đề sau:

* Làm việc với những điểm kiểm chứng, một sơ đồ và những sự biểu diễn bình thường.
* Sử dụng sự so sánh trong việc cập nhật những điểm kiểm chứng.
* Tạo một Funtional Test Project và một kịch bản báo cáo
* Chạy lại những kịch bản.
* Thực hiện kiểm thử hồi qui

**8. Creating a Functional Tester project.**

1. Trên Menu của phần mềm Rational Tester click **File** > **New** > **Functional Test Project.**
2. Đặt tên vào **Project Name** (ô này không được để trống).
3. Chọn đường dẫn trong **Project Location.**
4. Nếu trình điều khiển nguồn đã có sẵn, không cần chọn **Add the project to Source Control**.
5. Nếu việc kết hợp dự án đã có sẵn bạn không cần chọn **Associate the Functional Test Project with current Rational Project**.
6. Click **Finish** để hoàn thành.

**9. Recording a script (ghi lại một kịch bản)**

Trong phần này, bạn sẽ biết cách ghi lại một kịch bản bằng cách sử dụng Functional Tester Recording.

**Begin recording (Bắt đầu việc ghi lại).**

1. Để bắt đầu quá trình ghi lại, click nút Record a Functional Test Script () trên thanh Toolbar của phần mềm.



2. Trong truờng Script name đặt tên cho của trình ứng dụng mà bạn sẽ hực hiện.

3. Click Next.

4. Click Finish.

5. Trên thanh Toolbar click nút Monitor Message Preferences (). Bạn có thể sử dụng nhiều sự lựa chọn trong cùng thời gian với sự xuất hiện trên trình điều khiển của hộp văn bản trên màn hình.



6. Click Cancel.

7. Click nút Insert Script Support Commands () trên thanh Toolbar.



Cửa sổ Script Support Functions hiện ra, cho phép bạn gọi một kịch bản nữa, chèn vào một mục, một thời gian, hoặc chèn vào một lời chỉ dẫn, lời bình trong kịch bản của bạn.

8. Click Close.

**Start the application(Bắt đầu trình ứng dụng)**

1. Để bắt với quá trình kiểm thử trên trình ứng dụng, Click nút **Start Application** () trên thanh Toolbar.



2. Trong cửa sổ Start Application, chọn **ClassicsJavaA** và sau click **OK.**

**Recording actions(Sự hoạt động bảng ghi)**

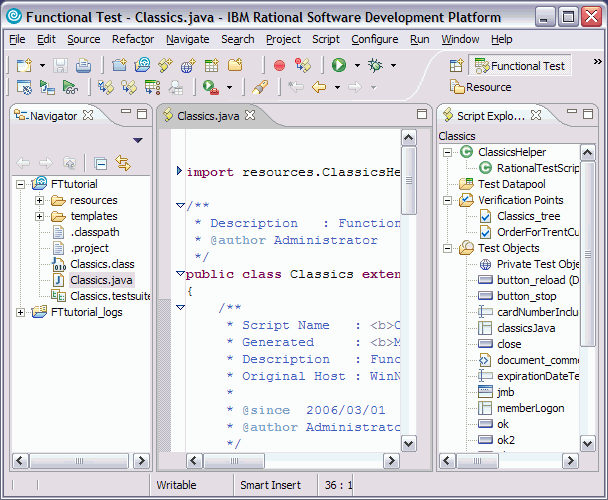
* + - 1. Click **­+** trên Foder mở rộng **Haydn** trên cây **Composers**.
      2. Hiển thị 1 List và bạ chọn **Symphonies Nos. 94 & 98**.
      3. Click nút **Place Order.**
      4. Trong cửa sổ Member Logon, để giữ sự hiển thị của **Existing Customer** and **Trent Culpito** bạn hãy Tick vào hộp Password.
      5. Click **Ok.**
      6. Trong hộp **Card Number** bạn đánh 777 777 777 777.
      7. Trong trường **expiration date**, bạn chọn 07/07.
      8. Click **Place Order.**
      9. Click **Ok**

**Playing back the script**

1. Để chạy lại kịch bản, Click nút **Run Functional Test Script** () trên thanh Toolbar



1. Trên cửa sổ Select Log, giữ nguyên sự hiển thị của bảng **Classics**, và chọn **Finish**.
2. Sau đó nó sẽ hiện cửa sổ như bên dưới.



4. Phía bên trái của cửa sổ là khung nhìn Funtional Tester Project. Ở đây hiển thị những dự án kiểm thử chức năng những dự án này hiện thời đang được kết nối. Tất cả những dự án trong khoảng thời gian kiểm thử này đều được liệt kê bên dưới với những cái tên của nó. Những dự án này cung cấp cho bạn những đường dẫn tới những kịch bản khác nhau. Khi bạn Click-double vào một trong những dự án đó thì cửa sổ hoạt động của kịch bản sẽ được mở ra.

Phía bên phải là một Scrip Explorer, trong này nó sẽ hiển thị những điểm kiểm chứng, những sơ đồ mục tiêu của kịch bản đang hoạt động. Từ Scrip Explorer, bạn có thể cho hiển thị những điểm kiểm chứng hay những sơ đồ mục tiêu. Nó sẽ cho bạn những thông tin cụ thể hơn.

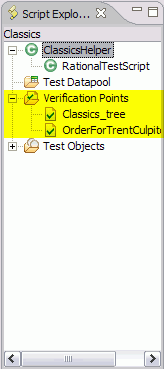
**10. Viewing verification points and object maps**

**Viewing verification points(Sự hiển thị của những điểm kiểm chứng.)**

Bạn có thể xem xét và sữa đổi dữ liệu trong những điểm kiểm chứng.

1. Trong Rational Funtional Tester, sẽ có sẵn một phần mềm là Classics.java giúp bạn chạy và tạo ra được những điểm kiểm chứng trong sự hoạt động của kịch bản.

2. Trong thư mục Verificational Points chứa 2 thư mục con nếu bạn muốn nó hiển thị thì hãy click vào đó.



3. Muốn hoàn tất quá trình bạn đóng Verificational Points

**Viewing object maps(Sự hiển thị của sơ đồ mục tiêu)**

1. Trong Scrip Explorer, chọn thư mục mở rộng **Test Objects**

2. Double-click **Private Test Object Map** () và nó sẽ được mở ra.



3. Phía trên đỉnh sẽ có dòng khai báo Frame: logFrame1: javax.swing.Jframe.

Cấu trúc của nó gồm hộp đăng nhập. Những nút lệnh, các trường password.

1. Click váo một biểu tượng.

Sự thừa nhận những đặc tính được hiển thị trên một cái lưới giống như dạng cây. Trong Menu Object Map bạn có thể thêm dữ liệu bằng cách click **Test Object** > **Insert Object(s)**.

2. Trong Menu Object Map, click **Preferences** > **Clear State On Close.**

Nút **Clear State On Close** là một nút chuyển hướng nó cho phép bạn bật tắc những hiển thị. Nếu như nó ở bên trái cửa sổ, tất cả những biểu tượng sẽ được thừa nhận khi bạn đóng sơ đồ và nó giúp chúng ta khi chúng ta muốn làm chậm quá trình quay lại khi chạy lại kịch bản.

**11.** **Performing** **regression tests**(Biểu diển kiểm thử hồ quy)

1. Trong Java Editor, kịch bản kiểm chứng Classics.java đang hoạt động

Trên đầu của kịch bản, dưới những thông tin khuôn mẫu, chú ý đến nút trên trình ứng dụng là

**startApp("ClassicsJavaA");**

2. Chuyển đổi A sang B.

3. Click nút **Run Functional Test Script** toolbar button ()



4. Trong cửa sổ Select Log, chọn **Classics** và chọn **Finish**.

5. Click **Yes**

**12. Using the Comparator to update a verification point?(Sự sử dụng dụng cụ so sanh để cập nhật một điểm kiểm chứng.)**

Bạn có thể sử dụng sự so sánh điểm kiểm chứng để so sánh dữ liệu của điểm kiểm chứng sau khi chạy lại kịch bản. Điểm kiểm chứng đưa ra môt dòng cơ bản về những đặc trưng hoặc dữ liệu liên quan của mục tiêu. Nếu việc xây dựng tiếp theo trên trình ứng dụng có những chỗ sai sót, bạn sẽ tìm được những sai sót đố và thậm chí còn tìm được những sự cố ý chuyển đổi trên trình ứng dụng. Nếu việc chuyển đổi là có chủ tâm, bạn có thể cập nhật những thông tin trên điểm kiểm chứng để cho việc kiểm thử tiếp theo xây dựng trong tương lai được đứng đắn.

* 1. Trên bảng Log, click **View Results** để hiểm thị những điểm kiểm chứng sai.

Khi một điểm kiểm chứng sai được hiển thị bạn so sánh kết quả mong đợi với bảng phân tích có gì khác nhau. Bạn có thể cho hiển thị những đường đường chỉ dẫn và hiệu chỉnh nó hoặc cập nhật nó với kết quả hiện có.

Kết quả không mong đợi sẽ được hiển thị với dòng chữ đỏ.

* 1. Cuộn **Text** lại.
  2. Click nút **Load Baseline to Edit** (). Chú ý vào sự hiển thị của cột **Baseline Value**.



* 1. Click nút **Jump to First Difference** () nút 4 hướng sẽ giúp bạn xác định sự khác nhau giữa những dòng cơ bản và những mẫu tin thực tế.



* 1. Click **Replace Baseline with actual value** button () đây là việc kiểm thử duy nhất chỉ sự khác nhau giữa phần phân tích và phần thực hiện.



* 1. Đóng ứng dụng lại.

Bây giờ chúng ta sẽ chạy lại kịch bản để kiểm chứng trở lại những điểm kiểm chứng bị lỗi.

* 1. Đóng bản Log.
  2. Click nút **Run Functional Test Script**.
  3. Chọn **Classics Log** và chọn **Finish.**
  4. Click **Yes.**
  5. Bản Log được phép mở ra.

**13. Updating the object map(Cập nhật sớ đồ mục tiêu)**

Trong phần này, bạn sẽ biết cách sữa những cảnh báo cho các đối tượng bằng cách sử dụng Object Map. Bạn sẽ sử dụng lặp đi lặp lại những biểu thức này một cách linh động nhất cho sự thừa nhận các đối tượng.

Khi bạn nhìn thấy những sự thừa nhận bị sai hoặc là có cảnh báo. Cuối phần II.2.5, bạn nhìn thấy phía trái bản Log mở ra, Click vào Project viwe chọn **Object Recognition is weak (above the warning threshold)**.

1. Nhìn vào trường **ObjectLookedFor** and **objectFound** trong đoạn cảnh báo gần dưới đáy của bảng Log.

Trong ClassicsA, chọn tên cho trường Password là **Remember Password.** Trong ClassicsB là **Remember The Password**. Khi bạn chạy lại kịch bản trong ClassicsB, sự thừa nhận của đối tượng sẽ không đúng đắn bởi vì nó khác nhau.

1. Nhìn vào trường **Line Number** trên bản Log. Chú ý đến số của nó và đóng bản Log để chạy lại Funtional Tester.
2. Click bất kỳ chỗ nào trên kịch bản, và sau đó click **Navigate** > **Go to Line**.
3. Phân loại kiểu số dòng từ hộp cảnh báo bị sai và sau đó click **OK**

Con trỏ sẽ di chuyển đến mép trái của số dòng

Bạn có thể tìm dòng bị lỗi bằng cách nhìn vào chỉ báo ở dưới đáy của cửa sổ Funtional Tester.

Ví Dụ: “43:9” tương đương với vị trí số 9 trên dòng 43.

Dòng bị lỗi trong kịch bản có dạng như

RememberPassword().clickToState(SELECTED);