

RESEARCH ARTICLE

## Working with IBM Rational Functional Tester: A Software Testing Tool

A. Nirmal Kumar<sup>1\*</sup> and B.G. Geetha<sup>2</sup>

<sup>1</sup>Dept. of Computer Science and Engineering, Christian College of Engineering and Technology, Dindigul-624619, TN;  
<sup>2</sup>Dept. of Computer Science and Engineering, KS Rangasamy College of Technology, Thiruchengode-637215, TN, India  
sa.nirmalkumar@gmail.com\*; +91 9443998771

### Abstract

In the software development life cycle, the testing phase only can ensure the quality of the software. The software system can be evaluated by means of both manual testing and automated testing. There are lot of automated testing tools available in this information technology revolution. Some of the automated software testing tools are Rational Functional Tester, WinRunner, LoadRunner, Quick Test Professional, OpenSTA, Test Director, SilkTest, Rational Robot, Silk Performer, Astra Load Test and Astra Quick Test. The automated software testing tool will be selected based on the nature of the software system or project and the experience level of tester to test the software. The process of testing the software by giving the manual inputs is called manual testing. The process of testing the software products with the help of software tools is called automated testing. This study presents about how the software is tested by Rational Functional Tester and also the working procedure of Rational Functional Tester is briefly explained. By installing IBM rational Functional tester properly, anyone can test the software which they have been developed by using the explained working procedure. While comparing with the other tools, rational functional tester tool will generate the test cases efficiently to test the software.

**Keywords:** Testing phase, manual testing, automated testing tools, manual inputs, rational functional tester.

### Introduction

Software testing is the process of evaluating, checking or ensuring the quality of software (Anisha Kumar, 2009). A product should be tested for quality after its manufacture. In software testing, the expected results are checked with the actual results (Rattan, 2010). If there is a difference between actual and expected results, the testers will conclude that there is a defect in the software products (Cullyer and Storey, 1994). If both the actual results obtained and expected results are equal, then the tester will assure the quality of software (Poulding and Clark, 2010). The software testing should be used to identify as well as reduce the errors (Kaur, 2011). The software industries are testing their developed software in two ways namely manual testing and automated tool testing (Mesbah *et al.*, 2012). Manual testing takes much more time relatively when compared to automated testing (Roper and Smith, 1987). The automated testing speeds up the testing process (Elbaum *et al.*, 2009). Manual testing requires a heavy investment in human resources. In automated testing, we can create test scripts that check all aspects of application. While testing the software, the test cases are generated. In manual testing, the test cases are generated by testers by their logical thinking (Luo and Ural, 1995). The software should be tested with all combinations of possible input values (Sturmer *et al.*, 2007). Hence, by using the manual testing, it is not at all possible to test the software with all the input values. But in case of automated testing, more number of test

cases is automatically generated within the short period of time (Kaur, 2013). The benefits of automated testing are fast, reliable, comprehensive and reusable (Mei *et al.*, 2012). The tools which are used only for testing is called as automated test tools (Pietrantuo and Trivedi, 2010). The test tools are classified into functional test tools, performance test tools and test management tools (Shahid, 2011). The functional test tools are used for checking the functionality of the softwares namely Rational Functional tester and WinRunner. In this study, Rational Functional Tester tool is taken as an example and the software is tested by Rational Functional Tester (RFT). The working procedure of RFT is briefly explained in this study and the steps to be followed or how to use this tool while working is also studied.

### Materials and methods

**Rational Functional Tester (RFT):** Rational Functional Tester is an open source tool that can be downloaded from internet and installed in computer. The installation steps are described below.

**Installation procedure:** Based on the user account (admin /non-admin), the installation will be varied. If user is admin then start the launchpad and click install RFT. If the user is non-admin then start the installation from the Setup disk by running the userinst program. If Installation Manager does not exist, then it will display the Packages. After the packages are installed, the Installation Manager will close and restart.

To display all the versions of package, click to show all the versions. We can check for other versions and extensions also. While installing, the internet access is also important. The installation manager will search for the repository. Give the license agreement properly. The path should be shared resource directory (e.g.: C:\Program Files\). Next, we should specify the package group. Then extend Eclipse IDE (Integrated Development Environment). We have to select the languages for package group. Next, features page will be implemented. Finally, when the installation process is completed, the confirm message will be displayed. After installing RFT, we can test the software by using the following procedures.

In this section, sample of four labs have been explained to test the software. Lab 1 explains about how to start with RFT and creating a functional test project.

### LAB 1: GETTING STARTED WITH IBM RATIONAL FUNCTIONAL TESTER:

View Functional Test Perspective:

Step 1: Click Start> Programs> IBM Software Delivery Platform> IBM Rational Functional Tester> Eclipse Integration.

Step 2: Select a workspace window will open. Browse to select the appropriate path to store the project and click ok.

Step 3: Close the welcome window once it open.

Step 4: If you are not already in the Functional Test perspective, click the Open a Perspective button on the right tab. (If the button is not displayed, click Window> Open Perspective >Other >Functional Test.).

Create a Functional Test Project:

Step 1: To create a project File->new->Functional Test Project.

Step 2: Functional Test Project window will open. Type project name as Functionaltesproj and click finish to create a project.

Help Contents:

Step 1: Click Help->Help contents.

Step 2: Expand Overview->Introduction to Rational Functional Tester. Practice navigating the Help topics.

Step 3: Close the Help window.

View in Functional Test Perspective:

Step 1: Go to Window-> Show view and Enable it.

Connect to an Existing Functional Test Project:

Step 1: Connect an Existing project by File->Connect to a functional Test Project.

Step 2: Click the Browse button to select the path of the existing file.

Step 3: Click Finish to connect to an existing functional test project.

Setting up the Test Environment

Enable the Environment for Testing:

Step 1: Go to Configure->Enable Environments for testing.

Step 2: In the Web Browsers environment Internet explorer is by default.

Step 3: In the Java Environments Java Runtime Environment is by default.

Configure the Application for Testing:

Step 1: Go to Configure->Configure Applications for Testing.

Thus, the functional test has been created by using the RFT. Lab 2 describes about recording the script with verification point and viewing the image verification point. These features are not available in other testing tools.

### LAB 2: RECORDING A SCRIPT

Record a Script:

Step 1: Start->programs->IBM Software Delivery platform->IBM Rational functional Tester->Eclipse Integration.

Step 2: Create a workspace as Recordwp and click ok.

Step 3: Create a project using File->new->Functional Test Project.

Step 4: Enter project name a Recordproj and click finish.

Step 5: Record a script using Record a functional Test script button (Red circle icon).

Step 6: Select a project as Recordproj and type the script name as placeorderscript1.

Step 7: Click next and by default private test object map is select in the test object map.

Step 8: Click finish to start recording.

Step 9: Recording monitor window will open. In that click start application icon to start the application.

Step 10: Select the application using Application name and click ok.

Step 11: Select a composer as Bach->violin concertos and then click place order button.

Step 12: Member log on window will open. Choose an existing customer as a Trent culpito.

Step 13: Enter credit card number as 1234123412341234 and Exp date as 12/12.

Step 14: Click place order button. Message will display that "Your order has been received".

Step 15: Close the window and stop recording using stop recording button.

Record a script with Data verification point:

Step 1: Record a script using Record a functional Test script button (Red circle icon).

Step 2: Select a project a Recordproj and type the script name as placeordervpscript2.

Step 3: Click next and by default private test object map is select in the test object map.

Step 4: Click finish to start recording.

Step 5: Recording monitor window will open. In the click start application icon to start the application.

Step 6: Select the application using Application name and click ok.

Step 7: Select a composer as Bach->violin concertos and then click place order button.

Step 8: Member log on window will open. Choose an existing customer as a Trent culpito.

Step 9: Enter credit card number as 1234123412341234 and Exp date as 12/12.

Step 10: In the Recording Monitor, Click the Insert Verification Point or Action Command button to record verification point.

Step 11: Select the object to test. Drag the Object Finder tool icon over the dollar amount displayed as the total for the order in the application and then release the mouse button. Click Next.

Step 12: Select an action window will open. Select Perform Data verification point and click next.

Step 13: In the Insert Verification Point Data Command page, change the Verification Point Name to OrderTotalAmount and Data Value as Label Visible Text. Click Next.

Step 14: In the Verification Point Data page, click Finish. Verification point is recorded.

Step 15: In the application, click the Place Order button.

Step 16: Message will display that "You order has been received".

Step 17: Close the window and stop recording using stop recording button.

View the Verification Point:

Step 1: Go to Script Explore->Expand verification point->double click the OrderTotalAmount.

Record a Script with Image verification point:

Step 1: Record a script using Record a functional Test script button (Red circle icon).

Step 2: Select a project as Recordproj and type the script name as placeordervpscript3.

Step 3: Click next and by default private test object map is select in the test object map.

Step 4: Click finish to start recording.

Step 5: Recording monitor window will open. In that click start application icon to start the application.

Step 6: Select the application using Application name and click ok.

Step 7: Select a composer as Beethoven->Symphony no.5.

Step 8: Image will be display.

Step 9: In the Recording Monitor, click the Insert Verification Point or Action Command button to record verification point.

Step 10: Select the object to test. Drag the Object Finder tool icon over the image and then release the mouse button. Click Next.

Step 11: Select an action window will open. Select Perform Image Verification point and click next.

Step 12: Type Verification point name as vpimage. Click next.

Step 13: Click finish to create a verification point.

View the Verification Point:

Step 1: Go to Script Explore->Expand verification point->double click the vpimage.

Script Support functions:

Call Script:

Step 1: Record a script using Record a functional Test script button (Red circle icon).

Step 2: Select a project as Recordproj and type the script name as callscript.

Step 3: Click next and by default private test object map is select in the test object map.

Step 4: Click finish to start recording.

Step 5: Recording monitor window will open. In that click Insert Script support command icon.

Step 6: Select Call Script and select placeorderscript1 and click Insert code and select Placeordervpscript2 and click Insert code.

Step 7: Click close.

Step 8: Stop recording using stop recording button.

Run the Script:

Step 1: Run the Functional Test script (Callscript) by using Run Functional script icon.

Step 2: Specify a log name to display log files.

Step 3: Click next and specify playback options window will open.

Step 4: By default click next.

Step 5: Click finish to start playback.

Sleep

Step 1: Record a script using Record a functional Test script button (Red circle icon).

Step 2: Select a project as Recordproj and type the script name as sleepscript.

Step 3: Click next and by default private test object map is select in the test object map.

Step 4: Click finish to start recording.

Step 5: Recording monitor window will open. In that click start application icon to start the application.

Step 6: Select the application using Application name and click ok.

Step 7: Select a composer as Bach->violin concertos and then click place order button.

Step 8: Member log on window will open. Choose an existing customer as a Trent culpito.

Step 9: Place the cursor in credit card number textbox.

Step 10: Open the Recording monitor window. In that click Insert Script support command icon.

Step 11: Select sleep and specify time as 3.0 seconds and click insert code.

Step 12: Click close.

Step 13: Enter credit card number as 1234123412341234.

Step 14: Place the cursor in Expdate and open the Recording monitor window. In that click insert Script support command icon.

Step 15: Select sleep and specify time as 3.0 seconds and click insert code.

Step 16: Click close.

Step 17: Enter Expdate as 10/10.

Step 19: Click Place order button. Message will display that "your order has been received".

Step 20: Close the window and stop recording using stop recording button.

Run the script

Step 1: Run the Functional Test Script (Sleepscript) by using Run Functional script icon.

Step 2: Specify a log name to display log files.

Step 3: Click next and specify playback options window will open.



Step 4: By default click next.  
Step 5: Click finish to start playback.

Lab 3 explains about script playback, viewing the results, viewing the specific log, editing the script and also running the script.

### LAB 3: PLAYBACK A SCRIPT AND VIEW RESULTS

Playback a script:

Step 1: Run the Functional Test script (placeorderscript1) by using Run Functional script icon.

Step 2: Specify a log name to display log files.

Step 3: click next and specify playback options window will open.

Step 4: By default click next.

Step 5: Click finish to start playback.

Step 6: Playback monitor will open.

Viewing Playback Results:

Step 1: When playback finishes, view the log. Scroll through the log to see all information.

Step 2: Close the log window.

View Specific log:

Step 1: In the Functional Test project view expand the Recordproj\_logs icon.

Step 2: In the list of logs double click the placeorderscript1 to display the log.

For Verification point Comparator

Record a script:

Step 1: Record a script using Record a functional Test script button (Red circle icon).

Step 2: Select a project as Recordproj and type the script name as placeordervpscript4.

Step 3: Click next and by default private test object map is select in the test object map.

Step 4: Click finish to start recording.

Step 5: Recording monitor window will open. In that click start application icon to start the application.

Step 6: Select the application using Application name (Classic JavaA) and click ok.

Step 7: Select a composer as Bach->violin concertos and then click place order button.

Step 8: Member log on window will open. Choose an existing customer as a Trent culpito.

Step 9: In the Recording Monitor, click the Insert Verification Point or Action Command button to record verification point.

Step 10: In select the object to test window, Drag the Object Finder tool icon over the Remember Password checkbox and release the mouse button. Click Next.

Step 11: Select an action window will open. Select Perform Data Verification point and click next.

Step 12: In the Data Value list, select Checkbox Visible Text, and then click next.

Step 13: Click finish to create a verification point.

Step 14: Click place order in the application. Place order window will open.

Step 15: Enter credit card number as 1234123412341234 and Exp date as 12/12.

Step 16: Click Place order button. Message will display that "Your order has been received".

Step 17: Close the window and stop recording using stop recording button.

Edit and Run The Script:

Step 1: Open the Placeordervpscript4.

Step 2: Instead of ClassicJavaA change it as ClassicJavaB in the script.

Step 3: Run the script with changes.

Step 4: Run the Functional Test Script (Placeordervpscript4) with changes by using Run Functional Script icon.

Step 5: Specify a different log name to display log files.

Step 6: Click next and specify playback options window will open.

Step 7: By default click next.

Step 8: Click finish to start playback.

View verification point in script Explorer:

Step 1: Go to Script Explorer->Expand verification point->double click the Rememberpassword (name of the verification point).

Step 2: Verification point Comparator will open.

Step 3: observe the value of the verification point Comparator.

Step 4: Close the Window.

Functional Test Preferences and options:

Step 1: On the Functional Test menu bar, click Window> Preferences.

Step 2: In the left pane, expand Functional Test by clicking the + sign.

Step 3: In the left pane, click Playback.

Step 4: The general playback settings are displayed in the right pane.

Step 5: Expand Playback by clicking the + sign and then click monitor.

Step 6: The playback monitor setting is displayed in the right pane.

Step 7: Clear the Show monitor during playback checkbox.

Step 8: In the left pane, click Logging.

Step 9: The logging potions are displayed in the right pane.

Step 10: Clear the Use Default checkbox, click the dropdown menu, and select text.

Step 11: Click OK. This closes the Preferences window.

Playback a script:

Step 1: Run the Functional Test Script (Placeorderscript1) by using Run Functional script icon.

Step 2: Specify a log name to display log files.

Step 3: Click next and specify playback options window will open.

Step 4: By default click next.

Step 5: Click finish to start playback.

Lab 4 describes about object recognition and Set Recognition Score Threshold. Also we can view the script in script explorer.



#### LAB 4: OBJECT RECOGNITION

Record a Script:

Step 1: Record a script using Record a functional Test script button (Red circle icon).

Step 2: Select a project as ObjectRecognproj and type the script name as placeordervpscript5.

Step 3: Click next and by default private test object map is select in the test object map.

Step 4: click finish to start recording.

Step 5: Recording monitor window will open. In that click start application icon to start the application.

Step 6: Select the application using Application name (Classic JavaA) and click ok.

Step 7: Select a composer as Bach->violin concertos and then click place order button.

Step 8: Member log on window will open. Choose an existing customer as a Trent culpito.

Step 9: In the Recording Monitor, click the Insert Verification Point or Action Command button to record verification point.

Step 10: In select the objects to test window, Drag the Object Finder tool icon over the Remember Password checkbox and release the mouse button. Click Next.

Step 11: Select an action window will open. Select Perform Data verification point and click next.

Step 12: In the Data Value list, select CheckBox Visible Text, and then click Next.

Step 13: Click finish to create a verification point.

Step 14: Click Place order in the application. Place order window will open.

Step 15: Enter credit card number as 1234123412341234 and Exp date as 12/12.

Step 16: Click place order button. Message will display that "Your order has been received".

Step 17: Close the window and stop recording using stop recording button.

Edit and Run The Script:

Step 1: Open the placeordervpscript5.

Step 2: Instead of ClassicJavaA change it as ClassicJavaB in the script.

Step 3: Run the script with changes.

Step 4: Run the Functional Test Script (placeordervpscript5) with changes by using Run Functional script icon.

Step 5: Specify a different log name to display log files.

Step 6: Click next and specify playback options window will open.

Step 7: By default click next.

Step 8: Click finish to start playback.

View verification point in script Explorer:

Step 1: Go to Script Explorer->Expand verification point->double click the Rememberpassword (name of the verification point).

Step 2: Verification point Comparator will open.

Step 3: Observe the value of the verification point Comparator.

Step 4: click update baseline with verification point icon.

Step 5: Close the window.

Playback the Script:

Step 1: Run the Functional Test Script (Placeordervpscript5) with changes by using Run Functional Script icon.

Step 2: Specify a different log name to display log files.

Step 3: Click next and specify playback options window will open.

Step 4: By default click next.

Step 5: Click finish to start playback.

View log Files:

Step 1: In the files there is a warning called Object Recognition is weak (above the warning threshold).

Step 2: There is a problem with Remember password object.

Set Recognition Score Threshold:

Step 1: Go to Windows->Preferences->Expand Functional test->expand playback.

Step 2: Click Script Assure and drag the warning level slider to no one.

Step 3: Playback the same script by using Run Functional script icon.

Step 4: log file will open.

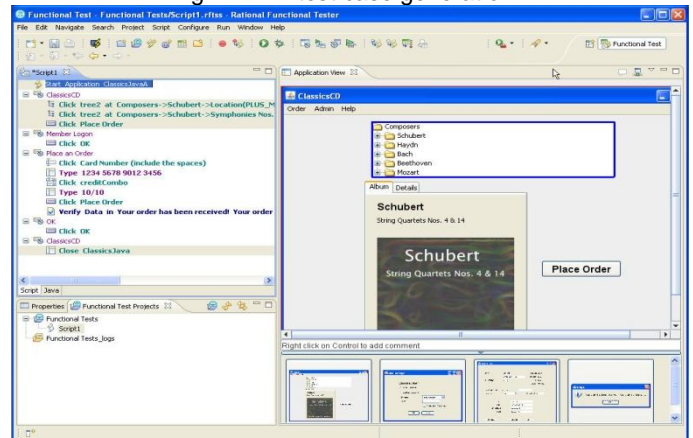
Step 5: Now no recognition warning will display in the log file.

By carefully following the above procedures, the developed software is tested with the help of RFT.

#### Results and discussion

The features of Rational Functional Tester (RFT) are recording script with data verification point, calling scripts, sleep functions, play backing a script, view specific log, verification point comparator, object recognition, view log files and setting recognition score threshold. The other advanced features are displaying a test object view, handling test object map, creating data driven test, handling data driven commands and handling data pools. These features are missing in other testing tools. Hence, Rational Functional Tester is the best software testing tools among the available automated software testing tools. While working on Rational Functional Tester, the following screenshots have been taken (Fig. 1).

Fig. 1. RFT test case generation.





From the screenshots results, we can work this tool. This tool has excellent graphical user interface and more user friendliness.

### Conclusion

By installing IBM rational Functional tester properly, anyone can test the software developed by using the above explained working procedure. While comparing with the other tools, rational functional tester tool will generate the test cases efficiently to test the software.

### Acknowledgements

Authors are very thankful to Dept. of Computer Science and Engineering, KS Rangasamy College of Technology, Tiruchengode, TN, India for their support to carry out this study.

### References

1. Anisha Kumar. 2009. Approaches of the WinRunner and LoadRunner software testing tools: A case study. *IUP J. Comp. Sci.* 3(4): 64-73.
2. Cullyer, V.J. and Storey, N. 1994. Tools and techniques for the testing of safety-critical software. *Comp. Control Engg. J.* 5(5): 239-244.
3. Elbaum, S., Chin, H.N., Dwyer, M.B. and Jorde, M. 2009. Carving and replaying differential unit test cases from system test cases. *IEEE Trans. Soft. Engg.* 35(1): 29-45.
4. Kaur, H. 2013. Comparative study of automated testing tools: Selenium, quick test professional and test complete. *Int. J. Engg. Res. Appl.* 3(5): 1739-1743.
5. Kaur, M. 2011. Comparative study of automated testing tools. *Int. J. Comp. Appl.* 24(1): 1-7.
6. Luo, G. and Ural, H. 1995. Approach to constructing software unit testing tools. *Soft. Engg. J.* 10(6): 245-252.
7. Mei, H., Hao, D. and Zhou, J. 2012. A static approach to prioritizing JUnit test cases. *IEEE Trans. Soft. Engg.* 38(6): 1258-1275.
8. Mesbah, A., Van Deursen, A. and Roest, D. 2012. Invariant-Based automated testing of modern web applications. *IEEE Trans. Soft. Engg.* 38(1): 35-53.
9. Pietrantuo, R. and Trivedi, K. 2010. Software reliability and testing time allocation: An architecture-based approach. *IEEE Trans. Soft. Engg.* 36(3): 323-337.
10. Poulding, S. and Clark, J.A. 2010. Efficient software verification: Statistical testing using automated search. *IEEE Trans. on Soft. Engg.* 36(6): 763-777.
11. Rattan, R. 2013. Performance evaluation and comparison of software testing tool. *Int. J. Inform. Comp. Technol.* 3(7): 711-716.
12. Roper, R.M.F. and Smith, P. 1987. A software tool for testing JSP designed programs. *Soft. Engg. J.* 2(2): 46-52.
13. Shahid, M. 2011. A study on test coverage in software testing. *Int. Conf. on Telecommunication Technol. Appl.* Vol. 5.
14. Sturmer, I. and Pepper, P. 2007. Systematic testing of model-based code generators. *IEEE Trans. Soft. Engg.* 33(9): 622-634.