Oracle Enterprise Manager
Oracle Database and Application Testing

Application Testing Suite Lab

Session S318966
Oracle Enterprise Manager 11g
Application Testing Suite 9.1
Hands-on Lab

Introduction to Enterprise Manager 11g

Oracle Enterprise Manager 11g is the centerpiece of Oracle's integrated IT management strategy, which rejects the notion of management as an after-thought. At Oracle, we design manageability into each product from the start, enabling Oracle Enterprise Manager to then serve as the integrator of manageability across the entire stack encompassing Oracle and non-Oracle technologies. Fueled by this unique vision, Oracle Enterprise Manager 11g has introduced business-driven IT management to help IT deliver greater business value through three highly differentiated capabilities:

Business-driven application management, which combines industry-leading capabilities in real user experience management, business transaction management and business service management to improve application users' productivity while enhancing business transaction availability
Integrated application-to-disk management, which provides deep management across the entire Oracle stack to reduce IT management complexity and eliminate disparate point tools
Integrated systems management and support, which utilizes industry-first technology bring support services into the IT management console; enabling proactive IT administration, increased application and system availability, and improved customer satisfaction

What is Oracle Application Testing Suite?

Oracle Application Testing Suite is a suite of products for automated functional testing, load testing and test management of Web, packaged and SOA-based applications. There are three separately licensed products in the Oracle Application Testing Suite:

Oracle Load Testing for automated load testing of Web applications, packaged applications and SOA Web Services.
Oracle Test Manager for documenting and managing the overall test process including test requirements, test cases and issues.

What is OpenScript?

OpenScript is the next generation scripting platform for the Application Testing Suite. OpenScript, a component of Oracle Functional Testing, allows you to create automated functional and regression testing scripts for your Web applications, Web Services and Oracle Applications including EBS, PeopleSoft, Siebel and more. OpenScript also allows you to create load test scripts to test the performance of these same applications using Oracle Load Testing. OpenScript combines an intuitive, graphical scripting interface with a flexible, Eclipse-based Java IDE to support the needs of advanced QA professionals and novice testers alike. OpenScript’s unique integrated scripting platform can help you address all your test automation needs.
This lab will demonstrate:

- Creating a web application test script
- Playing back the script and analyzing the results
- Parameterizing inputs
- Adding test cases to verify aspects of the application
- Creating a function library (OPTIONAL EXERCISE)

Please feel free to seek assistance from the instructor or Oracle Demo staff at any point in time.

Before we start taking you through the demonstration, please note the following:

- You will be given a virtual machine address to use for this lab. For ease of reference, you may want to write this information below:

  Virtual Machine Address: __________________________________________________________

  Username: ____________________
  Password: ____________________

- Start the lab by connecting to that virtual machine using Window Remote Desktop (Start - All Programs - Accessories - Remote Desktop Connection). Enter the virtual machine address to connect to and username and password.

- Oracle Demo staff will also provide you with the address for the web application you will be scripting against. When you record your script, you will enter that URL in the recording browser launched by OpenScript. For ease of reference, you may want to write this URL below:

  Application Address: __________________________________________________________
Create a New Functional Test Script

1. Launch OpenScript using the desktop shortcut.

2. At the Welcome page, click “Go to the OpenScript Workbench”. You will see the default user interface for the OpenScript workbench scripting interface. Note: If at any time you need to reset OpenScript to the default UI layout, you can select “View ➔ Reset Perspective”.

Page 3 of 38
3. Select File -- New

4. Select "Web" under the Functional Testing folder and click "Next"

5. Under “My Repositories” select “Default”
6. Specify a unique name for your script (i.e. MyScript1) and click "Finish"

**Record Into Your New Test Script**

1. Click the Record button to start recording the script
2. In the Firefox browser which launches, navigate to the application URL provided by your Oracle Demo staff.

3. Click “Login” under “Patient” (be sure you are logging in as a patient, not Administrator or Physician)

4. In the “Email” field, enter “oracle@oracle.com” and in the Password field, enter “oracle” and click Submit.
5. Click “Profile” to view the patient’s profile
6. Click “Logout”

7. Click the “Stop” button on the OpenScript toolbar to stop recording the script. Close the browser window. In the left-hand pane, expand the “Run” node of the script to view the script elements.

8. Click the “Save” button and save the newly recorded script

9. Click the “Play” button to play the newly-recorded script and verify successful script execution in the results view. (Note: after the script has fully played back, you can close the browser window if it remains open.)
10. Right-click the “Finish” node and select “Add – Other...”. Under the “Browser” tree node, select “Close Browser” and click OK.

Script Details and Verifying Results

1. Select the first step under the “Run” node of the script and click the Java Code tab at the bottom of the script window. This displays the Java code view of the script. Changes made to either the Java code or the tree view will be reflected in the other view.
2. You can maximize the script pane by either clicking the maximize icon in the upper-right hand side of the pane or double-clicking the script tab itself.

3. Restore the normal script pane and return to the tree view. We will now inspect the script playback results. Note: If at any time you need to reset OpenScript to the default UI layout, you can select “View ➔ Reset Perspective”.

4. In the Results view in the bottom pane of the OpenScript window, expand the script result nodes to display playback information for individual script components.
5. Select the first WaitForPage node in the results. Then, in the Details view in the top right pane, note the Comparison tab that allows you to compare individual playback results with the original recorded content.

6. The Compare combo-box at the top of the Details pane allows you to select different attributes to compare. For example, you can select comparison of the HTML content of the page.
Add a Databank To Your Test Script

1. Expand the steps in your script and find the “setText” command where the Login value (oracle@oracle.com) was entered.
2. Right-click on the setText node and select Properties

3. Click the “Substitute Variable”: button to the right of the Value field
4. Select “Add New Databank”: and click Next

5. Under “My Repositories/Default” select the DataBank directory
6. Select the "avitek.csv" file in the DataBank directory. Click Next.

7. Verify the "Username" column is selected, or select it in the "Column" pull-down, then click Finish.

8. Note the updated Value in the setText dialog which references the databank filename alias and column, then click OK.
9. Right-click on the `setPassword` node and select Properties.

10. Click the “Substitute Variable” button to the right of the Value field.
11. Select the "Password" field under the “avitek” Databanks node, then click Finish.

12. Note the updated Value in the setText dialog which references the databank filename alias and column, then click OK.

13. Click the “Play” button to play the databanked script and note the new username/password entered from the databank during playback and verify successful script execution in the results view.
Add a Test Case To Your Test Script

1. Expand the steps in your script and find the “WaitForPage” command where the patient successfully logged in.

2. In the right-hand Details pane, click the Browser tab to display the rendered HTML of the page.

3. In the Details pane, use the mouse to highlight the text “Successfully logged in!” Then, right-click and select “Add Text Matching Test”

Page 18 of 38
4. Assign the test a name (“LoginSuccess”) and click OK.
5. Verify test functions properly by playing back the script.
Optional Exercise

For those with time to explore OpenScript further, you can complete the following exercise to create a function library that you can use in other scripts. Then, create a driver script to use those functions.

1. Close all open scripts, then select File – New
2. Select "Web" under the Functional Testing folder and click "Next"

7. Under “My Repositories” select "Default"

3. Specify a unique name for your function library (i.e. MyFunctions) and click "Finish"
4. **Right-click the Run node. Select Add -- Other...**

5. **Under Script Functions -- Local Script, select [New Function] and click OK**
6. Give the function a name (e.g., “patientLogin”) and click OK

7. Repeat steps 5 and 6 to create two additional functions (e.g., viewProfile and logout)

8. From the “Set Record Section” drop-down on the toolbar, change the record section to the “patientLogin” function
9. Click the Record button to start recording the script.

10. In the Firefox browser which launches, navigate to the application URL provided by your Oracle Demo staff.
11. Click “Login” under “Patient” (be sure you are logging in as a patient, not Administrator or Physician)

12. In the “Email” field, enter “oracle@oracle.com” and in the Password field, enter “oracle” and click Submit.
13. From the “Set Record Section” drop-down on the OpenScript toolbar, change the record section to the “viewProfile” function
14. Click “Profile” to view the patient’s profile

15. From the “Set Record Section” drop-down on the OpenScript toolbar, change the record section to the “logout” function
16. Click “Logout”

17. Click the “Stop” button on the OpenScript toolbar to stop recording the script. Close the browser window. You can expand the different functions of the script to view their respective elements.

18. Click the “Save” button and save the script.

19. Now select File – New
20. Select "Web" under the Functional Testing folder and click "Next"

21. Under “My Repositories” select "Default"

22. Specify a unique name for your script (i.e. MyDriverScript) and click "Finish"
23. Right-click the Initialize node and select Add – Other… -- Browser – Launch Browser and click OK

24. Right-click the Finish node and select Add – Other… -- Browser – Close Browser and click OK
25. Select Script – Script Properties…

26. Click on Script Assets
27. Select Scripts and click Add…

28. Scroll down to find MyFunctions
29. Click OK

30. Click OK on the Script Properties dialog
31. Right-click the Run node. Select Add – Other… – Script Functions – Script: MyFunctions – patientLogin

32. Click OK
33. Click OK on the Call Function dialog

34. Repeat steps 29-31 for script functions “viewProfile” and “logout” respectively

35. Click the “Save” button
This concludes the Oracle Enterprise Manager Hands-on Lab.

Conclusion

OpenScript is the next generation scripting platform for the Application Testing Suite. OpenScript, a component of Oracle Functional Testing, allows you to create automated functional and regression testing scripts for your Web applications, Web Services and Oracle Applications including EBS, PeopleSoft, Siebel and more. OpenScript also allows you to create load test scripts to test the performance of these same applications using Oracle Load Testing. OpenScript combines an intuitive, graphical scripting interface with a flexible, Eclipse-based Java IDE to support the needs of advanced QA professionals and novice testers alike. OpenScript’s unique integrated scripting platform can help you address all your test automation needs.

Key Features of OpenScript

Application Support:
- Support for Web application automated functional testing and load testing
- Support for Siebel application automated functional testing and load testing
- Support for Oracle EBS/Forms application automated functional testing and load testing
- Support for Web Services functional testing and load testing

General Features:
- Intuitive graphical scripting interface and wizards for creating automated functional test scripts and load test scripts
- Integrated Java IDE based on Eclipse for viewing, editing and debugging script code
- Integrated with Oracle Load Testing for executing load test scripts
- Integrated with Oracle Test Manager for executing automated test scripts
- Command line interface for executing test scripts from 3rd party tools
For additional information, visit:

Oracle Application Testing Suite
http://www.oracle.com/technology/products/oem/prod_focus/etest.html

Oracle Application Quality Management
http://www.oracle.com/enterprise_manager/application-quality-solutions.html

Oracle Enterprise Manager
http://www.oracle.com/enterprise_manager/index.html

OTN Discussion Forum for Application Testing Suite