

TPCT's
College of Engineering, Osmanabad

Laboratory Manual

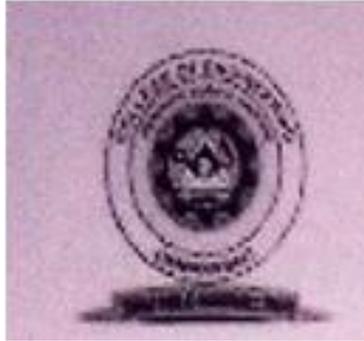
SOFTWARE TESTING AND QUALITY ASSURANCE

For Third Year Engineering Students

Manual Prepared by

Mr.R.A.Sarwade

Aauthor COE, Osmanabad



**TPCT's
College of Engineering
Solapur Road, Osmanabad
Department of Computer Science & Engineering.**

Vision of the Department:

To achieve and evolve as a center of academic excellence and research center in the field of Computer Science and Engineering. To develop computer engineers with necessary analytical ability and human values who can creatively design, implement a wide spectrum of computer systems for welfare of the society.

Mission of the Department:

The department strives to continuously engage in providing the students with in-depth understanding of fundamentals and practical training related to professional skills and their applications through effective Teaching- Learning Process and state of the art laboratories pertaining to CSE and inter disciplinary areas. Preparing students in developing research, design, entrepreneurial skills and employability capabilities.

College of Engineering

Technical Document

This technical document is a series of Laboratory manuals of **Computer Science & Engineering Department** and is a certified document of College of Engineering, Osmanabad. The care has been taken to make the document error-free. But still if any error is found. Kindly bring it to the notice of subject teacher and HOD.

Recommended by,

HOD

Approved by,

Principal

Copies:

1. Departmental Library
2. Laboratory
3. HOD
4. Principal

FOREWORD

It is my great pleasure to present this laboratory manual for Third year engineering students for the subject of Software testing and Quality Assurance.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relieve them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear..

H.O.D.

CSE Dept

LABORATORY MANUAL CONTENTS

This manual is intended for the Third year engineering students in the subject of Software Testing and Quality Assurance. This manual typically contains practical/lab sessions related to Software Testing and Quality Assurance implemented in Winrunner covering various aspects related the subject to enhanced understanding.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Mr. R.A.Sarwad

SUBJECT INDEX:-

1.Do's & Don'ts in Laboratory.

2. Lab Exercises

1. Introduction to winrunner.
2. Recording test in analog and context sensitive mode
3. Synchronizing test
4. Checking GUI Objects
5. Checking Bitmap Objects
6. Test with TSL
7. Creating data driven test
8. Manual Testing

3.Quiz

4.Conduction of viva voce examination

5.Evaluation & marking scheme

1.Do's and Don'ts in the laboratory

1. Make entry in the Log Book as soon as you enter the Laboratory.
2. All the students should sit according to their roll numbers starting from their left to right.
3. All the students are supposed to enter the terminal number in the log book.
4. Do not change the terminal on which you are working.
5. All the students are expected to get at least the algorithm of the program/concept to be implemented.
6. Strictly observe the instructions given by the teacher/Lab Instructor

Instruction for Laboratory Teachers::

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.
2. Students should be taught for taking the printouts under the observation of lab teacher

Pre-Lab

Pre-lab (Introduction to Win runner)

Questions:

- 1] What is Software Testing Tools?
 - 2] What is mean by Win Runner Testing Tools?
 - 3] **Exploring the Win Runner Window.**
- .

EXPERIMENT NO. 1

Title: Introduction to winrunner.

Objective: Student should be able to

- Describes the benefits of automated testing
- Understand the WinRunner testing process
- Work with WinRunner user interface

Theory:

Understanding the Testing Process

The WinRunner testing process consists of 6 main phases:

1 Teaching WinRunner the objects in your application

WinRunner must learn to recognize the objects in your application in order to run tests.

The preferred way to teach WinRunner your objects depends on the GUI map mode you select.

The two GUI map modes are described in detail in subsequent lessons.

2 Creating additional test scripts that test your application's functionality

WinRunner writes scripts automatically when you record actions on your application, or you can program directly in Mercury Interactive's Test Script Language (TSL).

3 Debugging the tests

You debug the tests to check that they operate smoothly and without interruption.

4 Running the tests on a new version of the application

You run the tests on a new version of the application in order to check the application's behavior.

5 Examining the test results

You examine the test results to pinpoint defects in the application.

6 Reporting defects

If you have the TestDirector 7.0i, the Web Defect Manager (TestDirector 6.0), or the Remote Defect Reporter (TestDirector 6.0), you can report any defects to a database. The Web Defect Manager and the Remote Defect Reporter are included in TestDirector, Mercury Interactive's software test management tool.

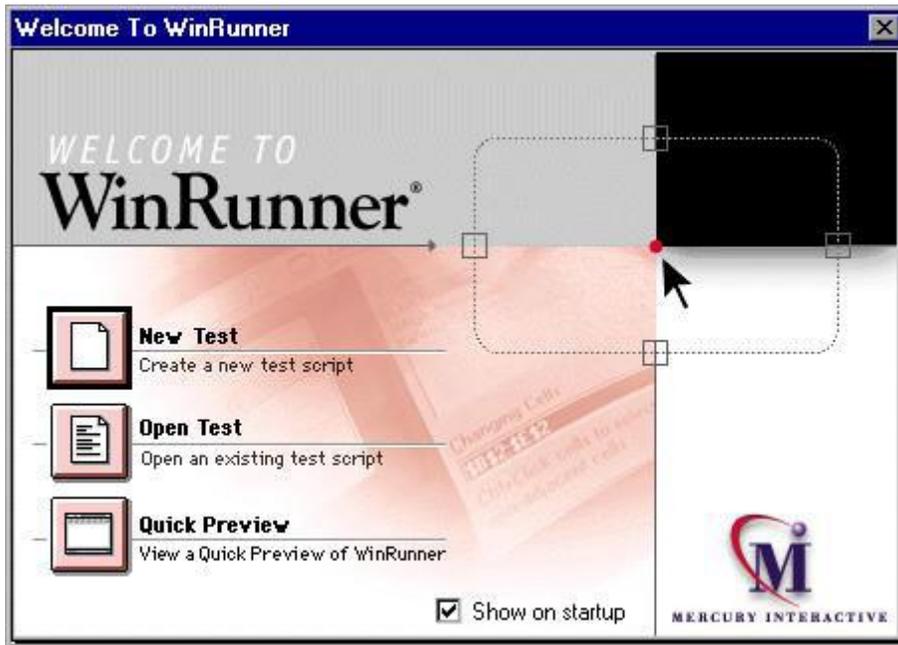
Exploring the WinRunner Window

Before you begin creating tests, you should familiarize yourself with the WinRunner main window.

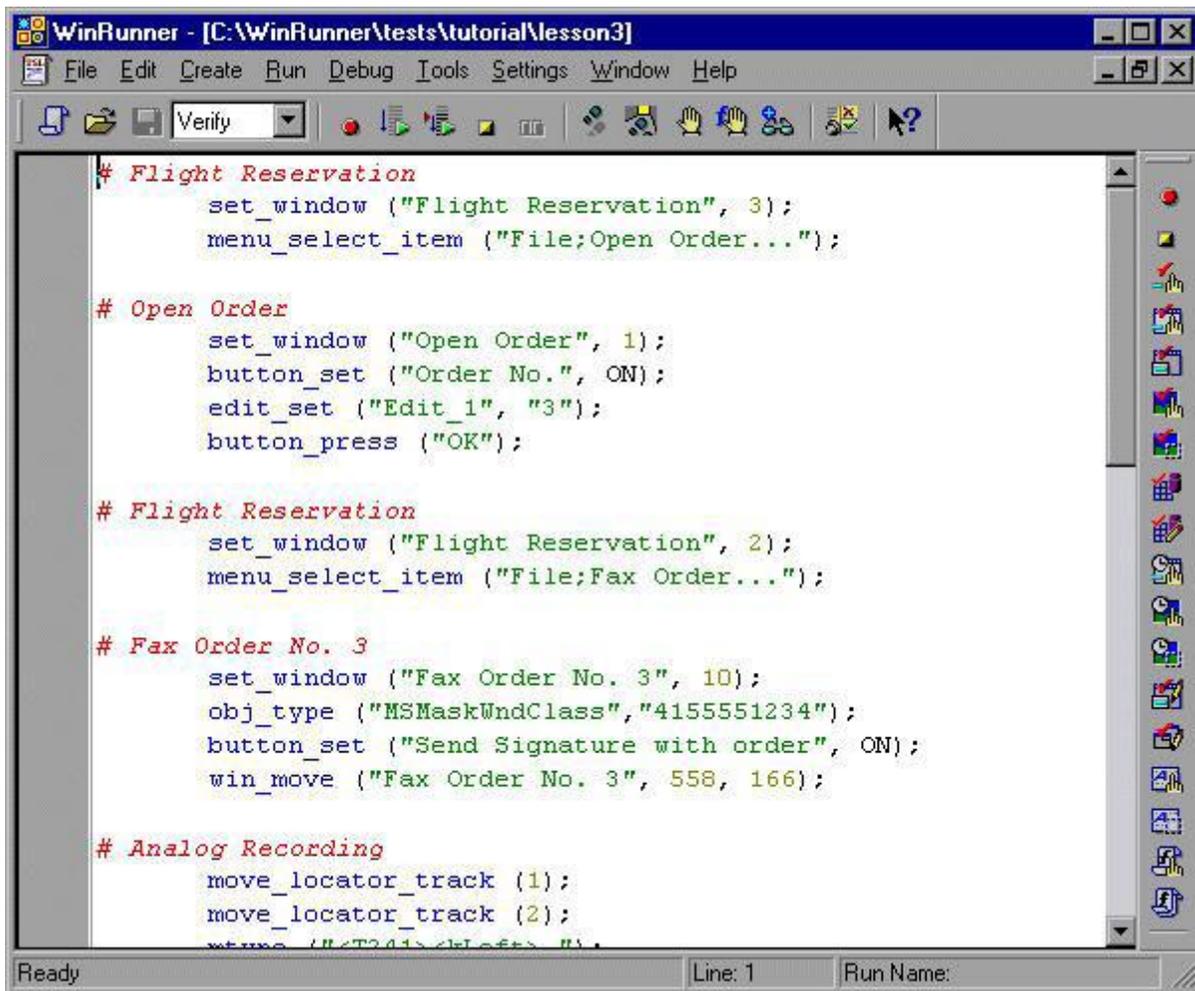
To start WinRunner:

Choose **Programs > WinRunner > WinRunner** on the **Start** menu.

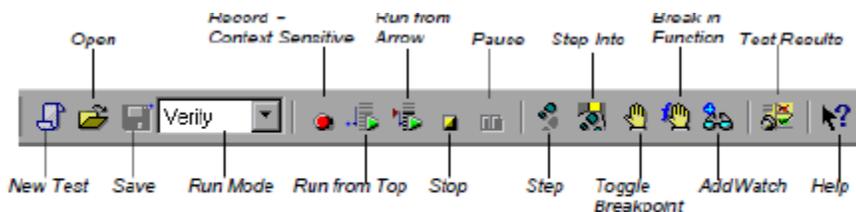
The first time you start WinRunner, the Welcome to WinRunner window opens. From the welcome window you can create a new test, open an existing test, or view an overview of WinRunner in your default browser.



The first time you select one of these options, the WinRunner main screen opens with the “What’s New in WinRunner” section of the help file on top. If you do not want the welcome window to appear the next time you start WinRunner, clear the **Show on startup** check box. Each test you create or run is displayed by WinRunner in a test window. You can open many tests at one time.



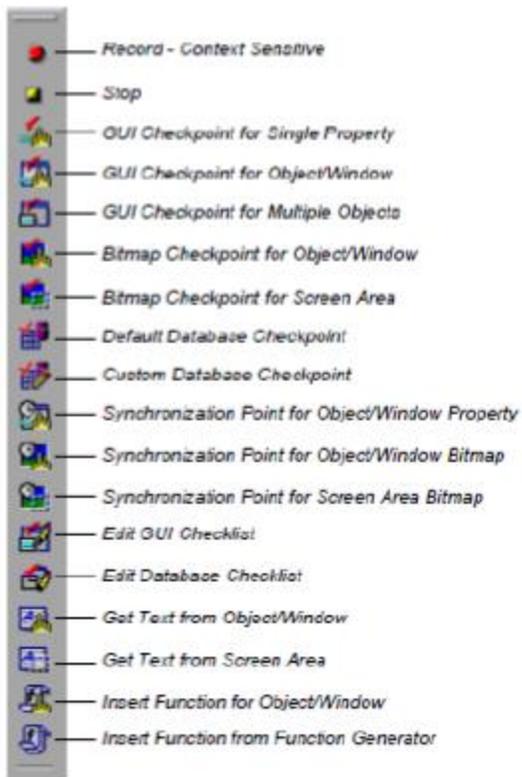
The *Standard toolbar* provides easy access to frequently performed tasks,



The *User toolbar* displays the tools you frequently use to create test scripts.

By default, the User toolbar is hidden.

To display the User toolbar choose **Window > User Toolbar**. When you create tests, you can minimize the WinRunner window and work exclusively from the toolbar.



The User toolbar is customizable. You choose to add or remove buttons using the **Settings > Customize User Toolbar** menu option. When you re-open WinRunner, the User toolbar appears as it was when you last closed it.

Experiment No. :2

Title: Recording test in analog and context sensitive mode

Objective: Student should be able to

- Describes Context Sensitive and Analog record modes
- Record a test script
- Read the test script
- Run the recorded test and analyze the results

Theory:

Choosing a Record Mode

By recording, you can quickly create automated test scripts. You work with your application as usual, clicking objects with the mouse and entering keyboard input.

WinRunner records your operations and generates statements in TSL, Mercury Interactive's Test Script Language. These statements appear as a script in a WinRunner test window.

Before you begin recording a test, you should plan the main stages of the test and select the appropriate record mode. Two record modes are available: Context Sensitive and Analog.

Context Sensitive

Context Sensitive mode records your operations in terms of the GUI objects in your application.

WinRunner identifies each object you click (such as a window, menu, list, or button), and the type of operation you perform (such as press, enable, move, or select).

For example, if you record a mouse click on the **OK** button in the Flight Reservation Login window, WinRunner records the following TSL statement in your test script:

```
button_press ("OK");
```

When you run the script, WinRunner reads the command, looks for the **OK** button, and presses it.

When choosing a record mode, consider the following points

Choose Context Sensitive if...	Choose Analog if...
The application contains GUI objects.	The application contains bitmap areas (such as a drawing area).
Exact mouse movements are not required.	Exact mouse movements are required.
You plan to reuse the test in different versions of the application.	

Recording a Context Sensitive Test:-

In this exercise you will create a script that tests the process of opening an order in the Flight Reservation application. You will create the script by recording in Context Sensitive mode.

- 1 Start WinRunner.**
- 2 Open a new test.**
- 3 Start the Flight Reservation application and log in.**
- 4 Start recording in Context Sensitive mode.**
- 5 Open order #3.**
- 6 Stop recording.**
- 7 Save the test.**

Recording in Analog Mode:-

In this exercise you will test the process of sending a fax. You will start recording in Context Sensitive mode, switch to Analog mode in order to add a signature to the fax, and then switch back to Context Sensitive mode

1 Open the Fax Order form and fill in a fax number.

2 Select the Send Signature with Order check box.

3 Sign the fax again in Analog mode.

4 Stop Recording.

5 Save the test.

Running the Test

You are now ready to run your recorded test script and to analyze the test results. WinRunner provides three modes for running tests. You select a mode from the toolbar.

Use *Verify mode* when running a test to check the behavior of your application, and when you want to save the test results.

Use *Debug mode* when you want to check that the test script runs smoothly without errors in syntax. See Lesson 7 for more information.

Use *Update mode* when you want to create new expected results for a GUI checkpoint or bitmap checkpoint. See Lessons 5 and 6 for more information

To run the test:

1 Check that WinRunner and the main window of the Flight Reservation application are open on your desktop.

2 Make sure that the *saved* test window is active in WinRunner.

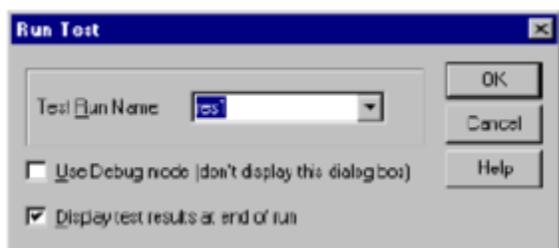
3 Make sure the main window of the Flight Reservation application is active.

4 Make sure that Verify mode is selected in the toolbar.

5 Choose Run from Top.

Choose **Run > Run from Top** or click the **Run from Top** button. The **Run Test**

dialog box opens.



6 Choose a Test Run name.

7 Run the test.

8 Review the test results.

Conclusion:-

WinRunner Results - H:\Program Files\Mercury
Interactive\WinRunner\tmp\noname14

=====

Expected results folder: H:\Program Files\Mercury
Interactive\WinRunner\tmp\noname14\exp
Test Results Name: H:\Program Files\Mercury
Interactive\WinRunner\tmp\noname14\res1
Operator Name:
Date: Tue Mar 17 12:00:21 2015
Summary:

Test Result: OK
Total number of bitmap checks: 0
Total number of GUI checks: 0
Total Run Time: 00:00:04
Detailed Results Description
Line Event Result Details Time

3 start run run noname14 00:00:00
23 stop run pass noname14 00:00:04
WinRunner Results - H:\Program Files\Mercury
Interactive\WinRunner\tmp\noname14

=====

Expected results folder: H:\Program Files\Mercury
Interactive\WinRunner\tmp\noname14\exp
Test Results Name: H:\Program Files\Mercury
Interactive\WinRunner\tmp\noname14\res1
Operator Name:
Date: Tue Mar 17 12:00:21 2015
Summary:

Test Result: OK
Total number of bitmap checks: 0
Total number of GUI checks: 0
Total Run Time: 00:00:04
Detailed Results Description
Line Event Result Details Time

3 start run run noname14 00:00:00
23 stop run pass noname14 00:00:04

Experiment No.: 3

Title: . To perform Synchronization test

Objective: Student should be able

- Describes when you should synchronize a test
- Synchronize a test
- Run the test and analyze the results

When Should You Synchronize?

When you run tests, your application may not always respond to input with the same speed. For example, it

might take a few seconds:

- to retrieve information from a database
- for a window to pop up
- for a progress bar to reach 100%
- for a status message to appear

WinRunner waits a set time interval for an application to respond to input. The default wait interval is up to 10

seconds. If the application responds slowly during a test run, WinRunner's default wait time may not be

sufficient, and the test run may unexpectedly fail.

If you discover a synchronization problem between the test and your application, you can either:

- Increase the default time that WinRunner waits. To do so, you change the value of the

Timeout for

Checkpoints and CS Statements option in the Run tab of the General Options dialog box (**Settings >**

General Options). This method affects all your tests and slows down many other Context Sensitive operations.

- Insert a *synchronization point* into the test script at the exact point where the problem occurs.

A

synchronization point tells WinRunner to pause the test run in order to wait for a specified response in

the application. *This is the recommended method for synchronizing a test with your application.*

In the following exercises you will:

- ✓ create a test that opens a new order in the Flight Reservation application and inserts the order into the database
- ✓ change the synchronization settings
- ✓ identify a synchronization problem
- ✓ synchronize the test
- ✓ run the synchronized test

Creating a Test

In this first exercise you will create a test that opens a new order in the Flight Reservation application and

inserts the order into a database

1 Start WinRunner and open a new test.

Start the Flight Reservation application and log in.

3 Start recording in Context Sensitive mode.

4 Create a new order.

5 Fill in flight and passenger information.

6 Insert the order into the database.

7 Delete the order.

8 Stop recording.

9 Save the test.

Changing the Synchronization Setting

1 Open the General Options dialog box.

2 Click the Run tab.

3 Change the value to 1000 milliseconds (1 second).

4 Click OK to close the dialog box.

Identifying a Synchronization Problem

1 Make sure that the *lesson4* test window is active in WinRunner.

2 Choose Run from Top.

3 Run the test.

4 Click Pause in the WinRunner message window.

Synchronizing the Test

1 Make sure that the *lesson4* test window is active in WinRunner.

2 Place the cursor at the point where you want to synchronize the test.

3 Synchronize the test so that it waits for the “Insert Done” message to appear in the status bar.

4 Manually change the 1 second wait in the script to a 10 second wait.

5 Save the test

Conclusion

WinRunner Results - H:\Documents and Settings\JNEC-12\Desktop\sytest

=====
=====

Expected results folder: H:\Documents and Settings\JNEC-12\Desktop\sytest\exp

Test Results Name: H:\Documents and Settings\JNEC-12\Desktop\sytest\res3

Operator Name: Date: Tue Mar 17 11:57:53 2015

Summary:

Test Result: OK

Total number of bitmap checks: 0

Total number of GUI checks: 0

Total Run Time: 00:00:07

Detailed Results Description

Line Event Result Details Time

3 start run run sytest 00:00:00

23 wait for bitmap OK Img1 00:00:07

31 stop run pass sytest 00:00:07

Experiment No.:4

Title: Checking GUI Objects.

Objective: Student should be able to

- Explain how to check the behavior of GUI objects
- Create a test that checks GUI objects
- Run the test on different versions of an application and examine the results

How Do You Check GUI Objects?

When working with an application, you can determine whether it is functioning properly according to the behavior of its GUI objects. If a GUI object does not respond to input as expected, a defect probably exists somewhere in the application's code. You check GUI objects by creating *GUI checkpoints*. A GUI checkpoint examines the behavior of an object's properties. For example, you can check: the content of a field whether a radio button is on or off whether a pushbutton is enabled or disabled

Adding GUI Checkpoints to a Test Script

- 1 Start WinRunner and open a new test.**
- 2 Start the Flight Reservation application and log in.**
- 3 Start recording in Context Sensitive mode.**
- 4 Open the Open Order dialog box.**
- 5 Create a GUI checkpoint for the Order No. check box.**
- 6 Enter "4" as the Order No.**
- 7 Create another GUI checkpoint for the Order No. check box.**
- 8 Create a GUI checkpoint for the Customer Name check box.**
- 9 Click OK in the Open Order dialog box to open the order.**
- 10 Stop recording.**
- 11 Save the test.**

Conclusion

WinRunner Results - H:\Program Files\Mercury Interactive\WinRunner\tmp\noname15

=====
=====
=====

Expected results folder: H:\Program Files\Mercury Interactive\WinRunner\tmp\noname15\exp

Test Results Name: H:\Program Files\Mercury Interactive\WinRunner\tmp\noname15\res2

Operator Name:

Date: Tue Mar 17 12:06:04 2015

Summary:

Test Result: OK

Total number of bitmap checks: 0

Total number of GUI checks: 3

Total Run Time: 00:00:02

Detailed Results Description

Line Event Result Details Time

3 start run run noname15 00:00:00
12 start GUI checkpoint--- gui1 00:00:01
12 end GUI checkpointOK gui1 00:00:01
23 start GUI checkpoint--- gui2 00:00:02
23 end GUI checkpointOK gui2 00:00:02
24 start GUI checkpoint--- gui3 00:00:02
24 end GUI checkpointOK gui3 00:00:02
25 stop run pass noname15 00:00:02

Experiment No.5

Title: . Checking Bitmap Objects

Objective: Student should be able to

- Explains how to check bitmap images in an application
- Create a test that checks bitmaps
- Run the test in order to compare bitmaps in different versions of an application
- Analyze the results

Theory:

How Do You Check a Bitmap?

If your application contains bitmap areas, such as drawings or graphs, you can check these areas using a *bitmap checkpoint*. A bitmap checkpoint compares captured bitmap images pixel by pixel.

Adding Bitmap Checkpoints to a Test Script

- 1 Start WinRunner and open a new test.
- 2 Start the Flight Reservation application and log in.
- 3 Start recording in Context Sensitive mode.
- 4 Open order #6.
- 5 Open the Fax Order dialog box.
- 6 Enter a 10-digit fax number in the Fax Number box.
- 7 Move the Fax Order dialog box.
- 8 Switch to Analog mode.
- 9 Sign your name in the Agent Signature box.
- 10 Switch back to Context Sensitive mode.
- 11 Insert a bitmap checkpoint that checks your signature.
- 12 Click the Clear Signature button.
- 13 Insert another bitmap checkpoint that checks the Agent Signature box.
- 14 Click the Cancel button on the Fax Order dialog box.
- 15 Stop recording.
- 16 Save the test.

Conclusion:

WinRunner Results - H:\Documents and Settings\JNEC-12\Desktop\bit

Expected results folder: H:\Documents and Settings\JNEC-12\Desktop\bit\exp

Test Results Name: H:\Documents and Settings\JNEC-12\Desktop\bit\res3

Operator Name:

Date: Tue Mar 17 11:44:42 2015

Summary:

Test Result: OK

Total number of bitmap checks: 2

Total number of GUI checks: 0

Total Run Time: 00:00:38

Detailed Results Description

Line Event Result Details Time

3 start run run bit 00:00:00

58 bitmap checkpoint OK Img1 00:00:37

67 bitmap checkpoint OK Img2 00:00:38

76 stop run pass bit 00:00:38

Experiment No.:6

Title: Testing with TSL.

Objective: Student should be able to

- Use visual programming to add functions to The recorded test scripts
- Add decisionmaking logic to a test script
- Debug a test script
- Run a test on a new version of an application and analyze the result

Steps:

1 Start WinRunner and open a new test.

2 Start the Flight Reservation application and log in.

3 Start recording in Context Sensitive mode.

4 Open order #4.

5 Open the Fax Order dialog box.

6 Click Cancel to close the dialog box.

7 Stop recording.

8 Save the test

9 Insert a blank line above the `button_press ("Cancel");` statement and place the cursor at the beginning of this line.

10 Open the Fax Order dialog box.

11 Query the # Tickets field

12 Query the Ticket Price field.

13 Query the Total field.

14 Close the Fax Order dialog box.

15 Save the test.

16 Place the cursor below the last `edit_get_text` statement in the *saved* script.

17 Add the following statements to the test script exactly as they appear below. Note that the tabs or spaces at the beginning of the second and fourth lines are optional.

```
if (tickets*price == total)
```

```
tl_step ("total", 0, "Total is correct.");
```

```
else
```

```
tl_step ("total", 1, "Total is incorrect.");
```

these statements mean: "If *tickets* multiplied by *price* equals *total*, report that the total is correct, otherwise (else) report that the total is incorrect.

18 Save the test.

Conclusion:

WinRunner Results - H:\Documents and Settings\JNEC-12\Desktop\tsl

=====

Expected results folder: H:\Documents and Settings\JNEC-12\Desktop\tsl\exp

Test Results Name: H:\Documents and Settings\JNEC-12\Desktop\tsl\res1

Operator Name:

Date: Fri Mar 27 16:14:44 2015

Summary:

Test Result: OK

Total number of bitmap checks: 0

Total number of GUI checks: 0

Total Run Time: 00:00:03

Detailed Results Description

Line Event Result Details Time

3 start run run tsl 00:00:00

26 tl_step --- Step: total, Status: Pass, Description: Total is correct00:00:03

30 stop run pass tsl 00:00:03

Experiment No.:7

Title: Creating data driven test

Objective: Student should be able to

- Use the DataDriver Wizard to create a data-driven test
- Use regular expressions for GUI object names that vary with each iteration of a test
- Run a test with several iterations and analyze the results.

Theory:

Once you have successfully debugged and run your test, you may want to see how the same test performs with multiple sets of data. To do this, you convert your test to a data-driven test and create a corresponding data table with the sets of data you want to test. Converting your test to a data-driven test involves the following steps:

- Adding statements to your script that open and close the data table.
- Adding statements and functions to your test so that it will read from the data table and run in a loop while it applies each set of data.
- Replacing fixed values in recorded statements and checkpoint statements with parameters, known as *parameterizing* the test. You can convert your test to a data-driven test using the DataDriver Wizard or you can modify your script manually. When you run your data-driven test, WinRunner runs the parameterized part(s) of the test one time (called an *iteration*) for each set of data in the data table, and then displays the results for all of the iterations in a single Test Results window.

In Lesson 7 you created a test that opened a specific flight order and read the number of tickets, price per ticket, and total price from a fax order dialog box in order to check that the total price was correct. In this lesson you will create a test that performs the same check on several flight orders in order to check that your application computes the correct price for various quantities and prices of tickets.

- 1 Create a new test from the test experiment 6.**
- 2 Run the DataDriver Wizard.**
- 3 Create a data table for the test.**
- 4 Assign a table variable name.**
- 5 Select global parameterization options.**
- 6 Select the data to parameterize.**
- 7 Open the data table.**
- 8 Add data to the table.**
- 9 Save and close the table.**
- 10 Save the test.**
- 11 Locate the Fax Order window in the *flight1a.gui* GUI map file.**
- 12 Modify the window label with a regular expression.**
- 13 Close the Modify dialog box.**
- 14 Modify the `tl_step` statements.**

Locate the first `tl_step` statement in your script. Delete the words “total is correct.” and replace them with, “Correct. "tickets" tickets at \$"price" cost \$"total".”

```
tl_step("total",0, "Correct. "tickets" tickets at $"price" cost $"total".");
```

Use the same logic to modify the next `tl_step` statement to report an incorrect result. For example:

```
tl_step("total", 1, "Error! "tickets" tickets at $"price" does not equal $"total".");
```

Now you will be able to see which data is used in each iteration when you view the results.

- 15 Save the test.**

Conclusion

WinRunner Test Results - [D:\Program Files\Morcury Interactive\WinRunner\Atap\Lesson8\Lesson8]

File Options Tools Window

ref6

Lesson8

Test Result: OK

- Total number of bitmap checkpoints: 0
- Total number of GUI checkpoints: 0

General Information

Line	Event	Detail	Result	Time
1	start run	Lesson8	run	00:00:00
22	U_step	Step total, Status: Pass, Description: Connect 4 tickets at \$329.40 cost \$1293.60	---	00:00:02
22	U_step	Step total, Status: Pass, Description: Connect 1 tickets at \$312.00 cost \$312.00	---	00:00:05
22	U_step	Step total, Status: Pass, Description: Connect 1 tickets at \$337.40 cost \$337.40	---	00:00:07
22	U_step	Step total, Status: Pass, Description: Connect 2 tickets at \$354.54 cost \$709.08	---	00:00:09
22	U_step	Step total, Status: Pass, Description: Connect 2 tickets at \$161.60 cost \$323.60	---	00:00:12
28	stop run	Lesson8	pass	00:00:12

Ready

Experiment No.: 8

Title: Manual Testing

This experiment helps you write manual Testcases for Login Form as below

TEST CASE ID	TEST DESCRIPTION	TEST PREREQUISITE	TEST INPUTS	TEST RESULTS
1.	USERNAME	Should precede with capital letter followed by small case	Jayshri	Accepted
2.	USERNAME	Should contain only alphabets	Jay123	Error
3.	USERNAME	Special characters not allowed	#jay	Error
4.	USERNAME	Should not precede with digits	123jaydp	Error
5.	USERNAME	Blank space or tab not allowed	Cidco aurangabad	Error
6.	PASSWORD	It should contain minimum 8 characters	cidcoabd	Accepted
7.	PASSWORD	Combination of digits and alphabets with special characters allowed	cidcoabd123#\$	Accepted
8.	PASSWORD	Should not precede with special characters	@cidcoabd	Error
9.	PASSWORD	One digit and one alphabet is compulsory	cidcoabd123#\$	Accepted
10.	PASSWORD	Should not exceed 16 characters	Ac3ysj2#\$%l@0klef\$	Error
11.	SUBMIT	Click once to login	Single click	Accepted
12.	SUBMIT	Double click ,no action will be	Double click	No action

13.	SUBMIT	Moving the cursor towards button , it gets highlighted	Move the cursor on button	Button highlighted
14.	CANCEL	Single click ,login gets cancelled	Single click	Cancelled
15.	CANCEL	Double click,no action will be performed	Double click	No action
16.	CANCEL	Moving the cursor on button,highlights it	Move the cursor on button	highlighted

3. Quiz on the Subject

1) What is the difference between the QA and software testing?

The role of QA (Quality Assurance) is to monitor the quality of the process to produce a quality of a product. While the software testing, is the process of ensuring the final product and check the functionality of final product and to see whether the final product meets the user's requirement.

2) What is Testware?

Testware is the subset of software, which helps in performing the testing of application. It is a term given to the combination of software application and utilities which is required for testing a software package.

3) What is the difference between build and release?

Build: It is a number given to Installable software that is given to testing team by the development team.

Release: It is a number given to Installable software that is handed over to customer by the tester or developer.

4) What are the automation challenges that QA team faces while testing?

Exploitation of automation tool

Frequency of use of test case

Reusability of Automation script

Adaptability of test case for automation

5) What is bug leakage and bug release?

Bug release is when software or an application is handed over to the testing team knowing that the defect is present in a release. During this the priority and severity of bug is low, as bug can be removed before the final handover. Bug leakage is something, when the bug is discovered by the end users or customer, and missed by the testing team to detect, while testing the software

6) What is data driven testing?

Data driven testing is an automation testing part, which tests the output or input values. These values are read directly from the data files. The data files may include csv files, excel files, data pools and many more. It is performed when the values are changing by the time.

7) Explain the steps for Bug Cycle?

Once the bug is identified by the tester, it is assigned to the development manager in open status. If the bug is a valid defect the development team will fix it and if it is not a valid defect, the defect will be ignored and marked as rejected. The next step will be to check whether it is in scope, if it happens so that, the bug is not the part of the current release then the defects are postponed. If the defect or bug is raised earlier then the tester will be assigned a DUPLICATE status. When bug is assigned to developer to fix, it will be given an IN-PROGRESS status. Once the defect is repaired, the status will be changed to FIXED. At the end the tester will give CLOSED status if it passes the final test.

8) What does the test strategy include?

The test strategy includes introduction, resource, scope and schedule for test activities, test tools, test priorities, test planning and the types of test that has to be performed.

9) Mention the different types of software testing?

Unit testing
Integration testing and regression testing
Shakeout testing
Smoke testing
Functional testing
Performance testing
White box and Black box testing
Alpha and Beta testing
Load testing and stress testing
System testing.

10) What is branch testing and what is boundary testing?

The testing of all the branches of the application, which is tested once, is known as branch testing. While the testing, which is focused on the limit conditions of the software is known as boundary testing.

11) What are the contents in test plans and test cases?

Testing objectives
Testing scope
Testing the frame
The environment
Reason for testing
The criteria for entrance and exit
Deliverables
Risk factors

12) What is Agile testing and what is the importance of Agile testing?

Agile testing is software testing, which involves the testing of the software from the customer point of view. The importance of this testing is that, unlike normal testing process, this testing does not wait for development team to complete the coding first and then doing testing. The coding and testing both goes simultaneously. It requires continuous customer interaction. It works on SDLC (Systems Development Life Cycle) methodologies, it means that the task is divided into different segments and compiled at the end of the task.

13) What is Test case?

Test case is a specific term that is used to test a specific element. It has information of test steps, prerequisites, test environment and outputs.

14) What is the strategy for Automation Test Plan?

The strategy for Automation Test Plan

Preparation of Automation Test Plan

Recording the scenario

Error handler incorporation

Script enhancement by inserting check points and looping constructs

Debugging the script and fixing the issues

Rerunning the script

Reporting the result

15) What is quality audit?

The systematic and independent examination for determining the quality of activities is known as quality audit. It allows the cross check for the planned arrangements, whether they are properly implemented or not.

16) How does a server or client environment affect software testing?

As the dependencies on the clients are more, the client or server applications are complex. The testing needs are extensive as servers, communications and hardware are interdependent. Integration and system testing is also for a limited period of time.

18) Explain stress testing, load testing and volume testing?

Load Testing: Testing an application under heavy but expected load is known as Load Testing. Here, the load refers to the large volume of users, messages, requests, data, etc.

Stress Testing: When the load placed on the system is raised or accelerated beyond the normal range then it is known as Stress Testing.

Volume Testing: The process of checking the system, whether the system can handle the required amounts of data, user requests, etc. is known as Volume Testing.

19) What are the five common solutions for software development problems?

Setting up the requirements criteria, the requirements of a software should be complete, clear and agreed by all. The next thing is the realistic schedule like time for planning, designing, testing, fixing bugs and re-testing. Adequate testing, start the testing immediately after one or more modules development. Use rapid prototype during design phase so that it can be easy for customers to find what to expect. Use of group communication tools.

20) What is a 'USE' case and what does it include?

The document that describes, the user action and system response, for a particular functionality is known as USE case. It includes revision history, table of contents, flow of events, cover page, special requirements, pre-conditions and post-conditions.

21) What is CRUD testing and how to test CRUD?

CRUD testing is another name for Black Box testing. CRUD stands for Create, Read, Update and Delete.

22) What is validation and verification in software testing?

In verification, all the key aspects of software developments are taken in concern like code, specifications, requirements and document plans. Verification is done on the basis of four things list of issues, checklist, walkthroughs and inspection meetings. Following verification, validation is done, it involves actual testing, and all the verification aspects are checked thoroughly in validation..

23) Mention what are the types of documents in QA?

The types of documents in QA are

Requirement Document

Test Metrics

Test cases and Test plan

Task distribution flow chart

Transaction Mix

User profiles

Test log

User profiles

Test incident report

Test summary report

24) What does the software QA document should include?

Software QA document should include

Specifications

Designs

Business rules

Configurations

Code changes

Test plans

Test cases

Bug reports

User manuals, etc

25) Mention how validation activities should be conducted?

Validation activities should be conducted by following techniques Hire third party independent verification and validation Assign internal staff members that are not involved in validation and verification activities Independent evaluation

4. Conduction of VIVA-VOCE Examinations:

Teacher should conduct oral exams of the students with full preparation. Normally the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the student in the subject is tested. Oral Exams are to be conducted in co-cordial situation. Teachers taking oral exams should not have ill thoughts about each other & courtesies should be offered to each other in case of opinion, which should be critically suppressed in front of the students.

5.Evaluation and marking system:

Basic honesty in the evaluation and marking system is essential and in the process impartial nature of the evaluator is required in the exam system. It is a primary responsibility of the teacher to see that right students who really put their effort & intelligence are correctly awarded.

The marking pattern should be justifiable to the students without any ambiguity and teacher should see that students are faced with just circumstance.