IBM Software Group | Rational software

IBM Rational Functional Tester for Java

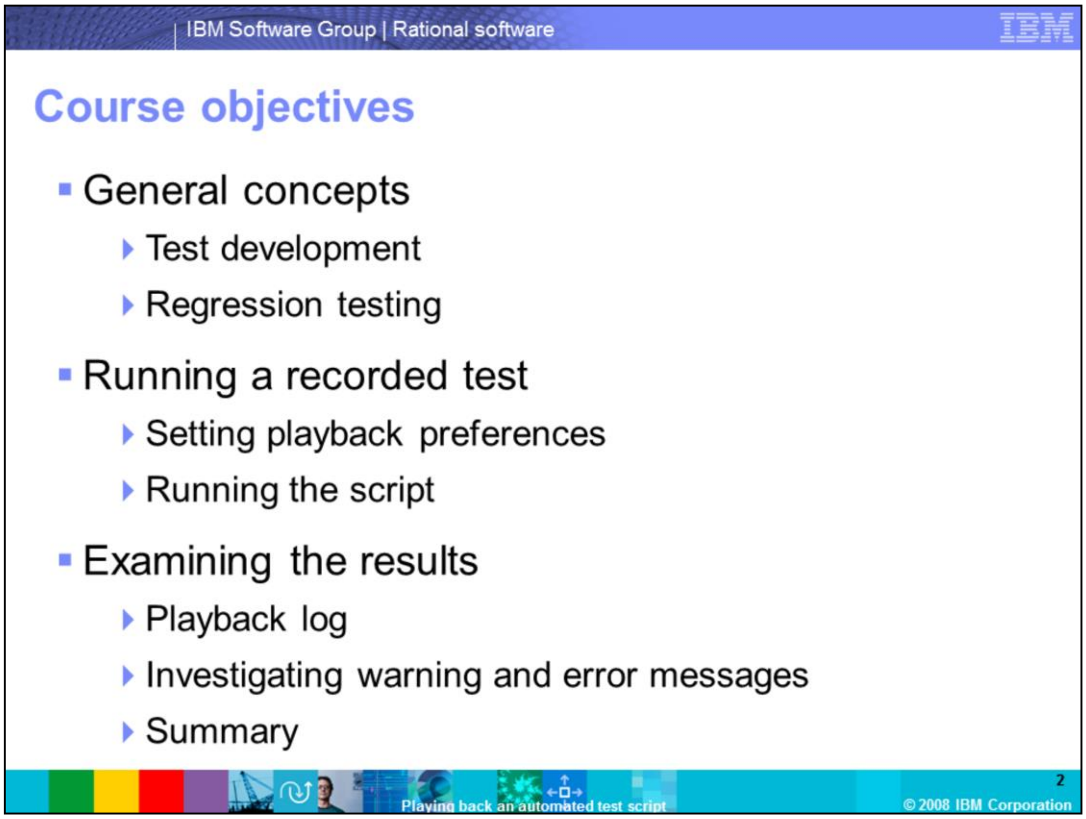
Playing back an automated test script

Rational software

e business on demand.

Converted to video May 22, 2015

This module provides an overview of playing back an automated test script using IBM Rational® Functional Tester for Java™.



IBM Software Group | Rational software

Course objectives

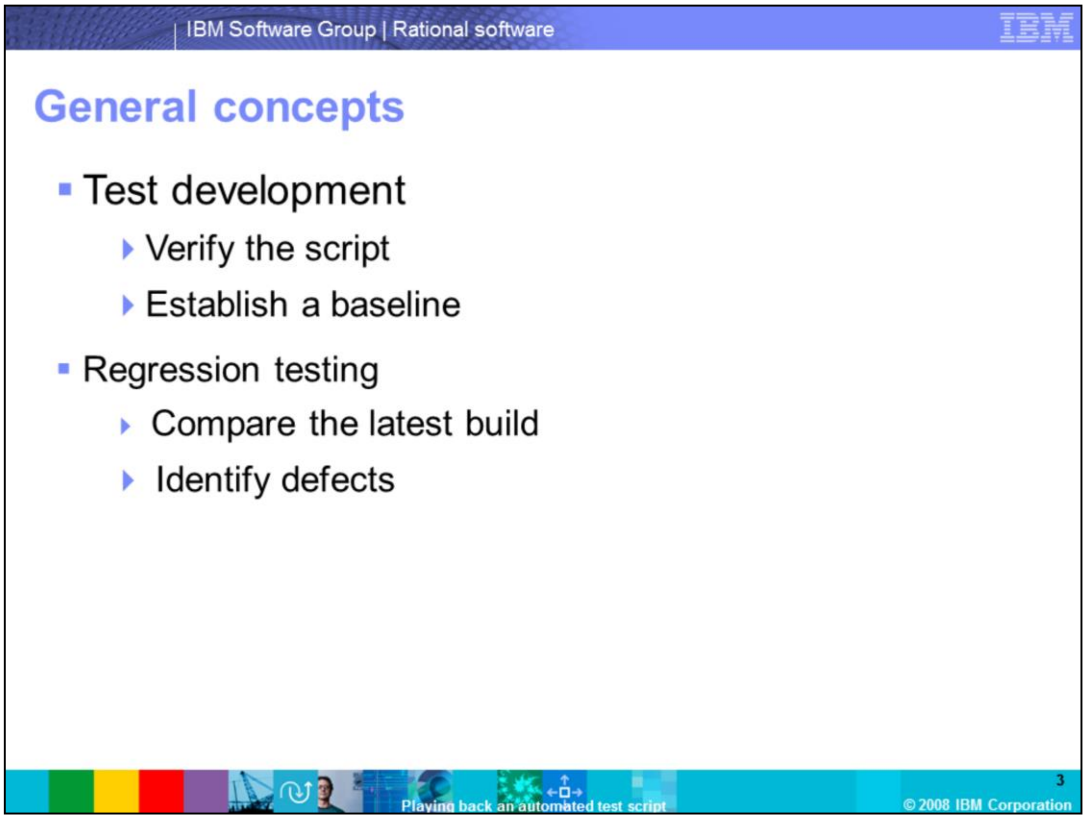
- General concepts
  - Test development
  - Regression testing
- Running a recorded test
  - Setting playback preferences
  - Running the script
- Examining the results
  - Playback log
  - Investigating warning and error messages
  - Summary

Playing back an automated test script

2

© 2008 IBM Corporation

This course covers topics related to playing back an automated test script. It discusses test development and regression testing, and goes through the steps of setting playback preferences and executing the script. This module also reviews logs, including warnings and errors in the logs.


IBM Software Group | Rational software

General concepts

- Test development
  - Verify the script
  - Establish a baseline
- Regression testing
  - Compare the latest build
  - Identify defects

Playing back an automated test script

3

© 2008 IBM Corporation

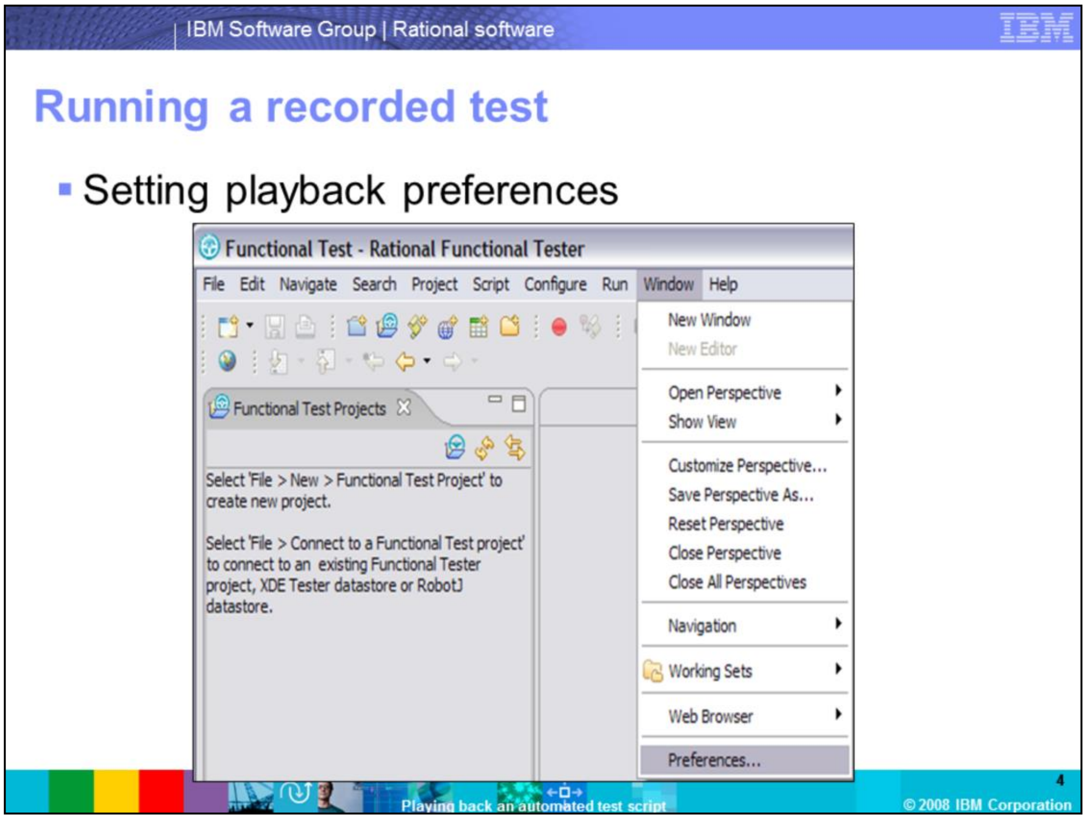
There are two phases of the automated test script development process that a script is played back in: test development and regression testing.

Test development includes recording, refining, and testing a script. After recording, a user can edit a script to add comments, perform additional actions, or use an external data source.

During test development, play back a script to verify that it plays back successfully and works as intended, and to establish a baseline against which to compare future builds of the application under test.

Regression testing reveals differences introduced into the application under test since the last build. Evaluate these differences to determine whether they are defects or deliberate changes.

During regression testing, play back a script to compare the latest build to an established baseline, and to identify defects caused to existing functionality by new code development.



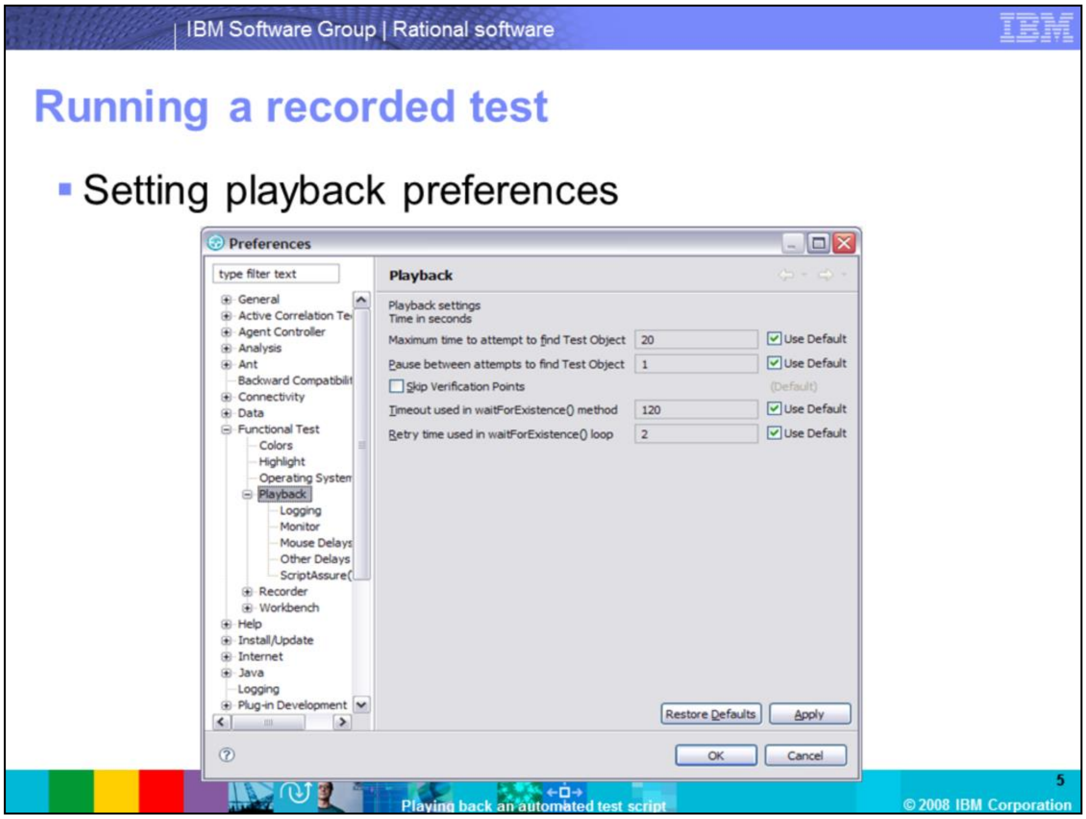
Running a recorded test

- Setting playback preferences

Select 'File > New > Functional Test Project' to create new project.

Select 'File > Connect to a Functional Test project' to connect to an existing Functional Tester project, XDE Tester datastore or RobotJ datastore.

Before playing back a recorded script, ensure that the application under test is in the same state it was in when the script was recorded. Any applications and windows that were open, displayed, or active when the script was recorded must be open, displayed, or active before playback is started.



Running a recorded test

- Setting playback preferences

Playback

Playback settings
Time in seconds

Maximum time to attempt to find Test Object	20	☑ Use Default
Pause between attempts to find Test Object	1	☑ Use Default
☐ Skip Verification Points		(Default)
Timeout used in waitForExistence() method	120	☑ Use Default
Retry time used in waitForExistence() loop	2	☑ Use Default

The tester has the option to set playback preferences. From the Window menu in Functional Tester, select “Preferences.” In the Preferences window that opens, in the left pane, expand “Functional Test,” then expand “Playback.”

Five different sets of playback preferences can be set here.

Logging allows testers to choose what kind of log, HTML, or Text they want to generate at the end of playback. Other available logging choices include “Prompt before overwriting an existing log,” “Display log viewer after script playback,” and “Don’t show script launch wizard.”

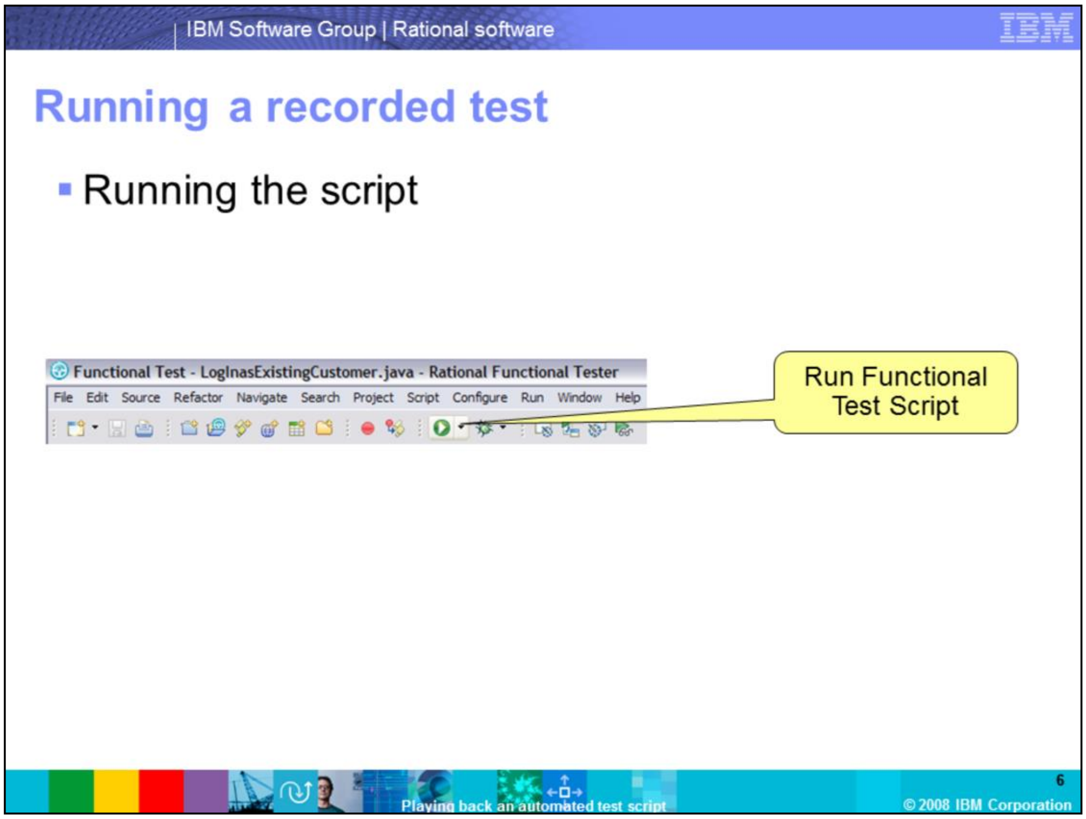
Monitor allows the tester to view the script name, the number of the line being executed, the status icons, and a description of the action in progress as the script is being played back.

Mouse Delay sets the delay in seconds to occur before every mouse up or down action.

Other Delays sets the delay in seconds to occur before every key up or down action, or before performing a test object action.

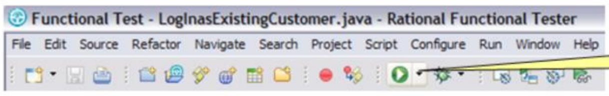
Script Assure helps a script to play back successfully, even after the update of the application under test. Each object in a test object map has a set of recognition properties, typically established during recording. If objects in the application under test have changed, scripts can still be played back in Functional Tester using the Script Assure feature to control object-matching sensitivity.

If the tester chooses to ignore these selections, the default settings are applied before playing back the script.

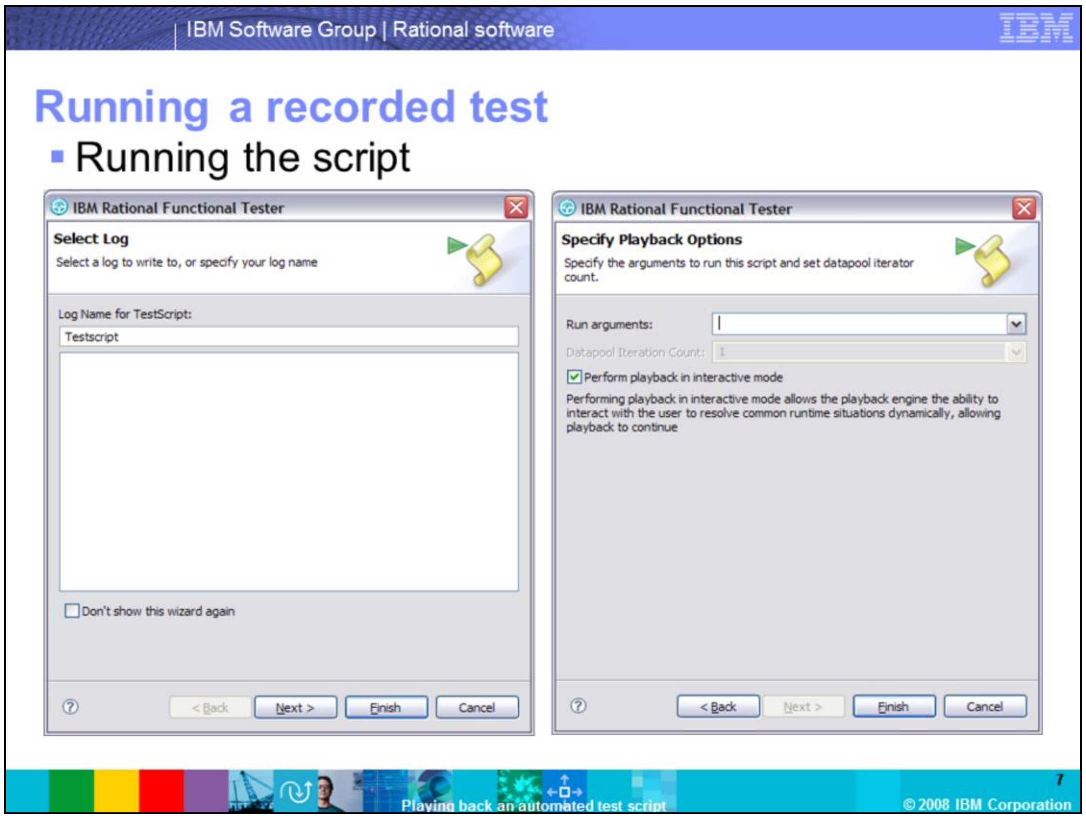


Running a recorded test

- Running the script


Functional Test - LogInasExistingCustomer.java - Rational Functional Tester
File Edit Source Refactor Navigate Search Project Script Configure Run Window Help
Run Functional Test Script

Once the playback preferences are set, the script is ready to execute. Open the script for playback, and click the “Run Functional Test Script” icon. A tester can also right-click the script in the Functional Test Project view, then click Run in the menu.



Running a recorded test

- Running the script

IBM Rational Functional Tester

Select Log

Select a log to write to, or specify your log name

Log Name for TestScript: Testscript

☐ Don't show this wizard again

IBM Rational Functional Tester

Specify Playback Options

Specify the arguments to run this script and set datapool iterator count.

Run arguments:

Datapool Iteration Count: 1

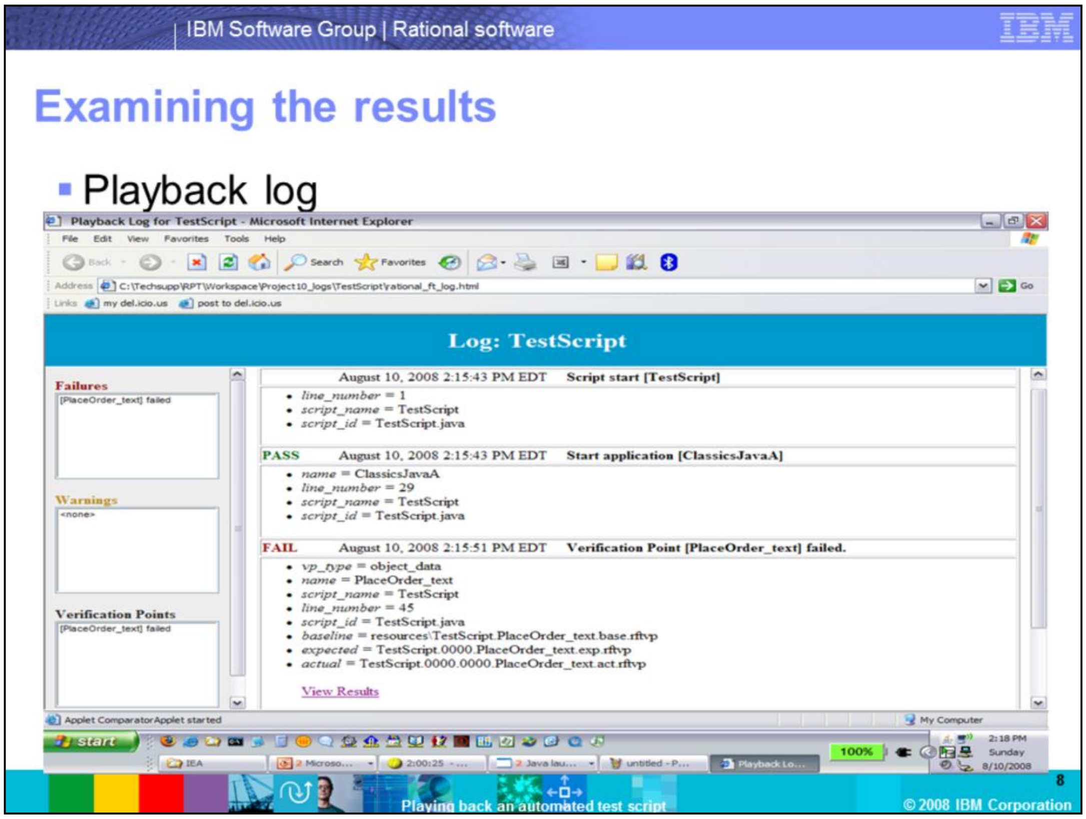
☑ Perform playback in interactive mode

Performing playback in interactive mode allows the playback engine the ability to interact with the user to resolve common runtime situations dynamically, allowing playback to continue

A Select Log window opens. By default, a log file named after the script appears. Testers have the option to rename this. Click **Next**.

A Specify Playback Options window opens. Here, testers can specify any arguments needed to run the script, and can set the datapool iteration count. When complete, click **Finish**.

Functional Tester now plays back the tests. Note that if this test was run previously, a prompt appears asking for confirmation to overwrite the log


IBM Software Group | Rational software

Examining the results

- Playback log

Playback Log for TestScript - Microsoft Internet Explorer

Address: C:\Techsupp\RPT\Workspace\Project10_logs\TestScript\rational_ft_log.html

Log: TestScript

Failures
[PlaceOrder_text] failed

Warnings
<none>

Verification Points
[PlaceOrder_text] failed

August 10, 2008 2:15:43 PM EDT **Script start [TestScript]**
- *line_number* = 1
- *script_name* = TestScript
- *script_id* = TestScript.java

PASS August 10, 2008 2:15:43 PM EDT **Start application [ClassicsJavaA]**
- *name* = ClassicsJavaA
- *line_number* = 29
- *script_name* = TestScript
- *script_id* = TestScript.java

FAIL August 10, 2008 2:15:51 PM EDT **Verification Point [PlaceOrder_text] failed.**
- *vp_type* = object_data
- *name* = PlaceOrder_text
- *script_name* = TestScript
- *line_number* = 45
- *script_id* = TestScript.java
- *baseline* = resources\TestScript.PlaceOrder_text.base.rftvp
- *expected* = TestScript.0000.PlaceOrder_text.exp.rftvp
- *actual* = TestScript.0000.0000.PlaceOrder_text.act.rftvp

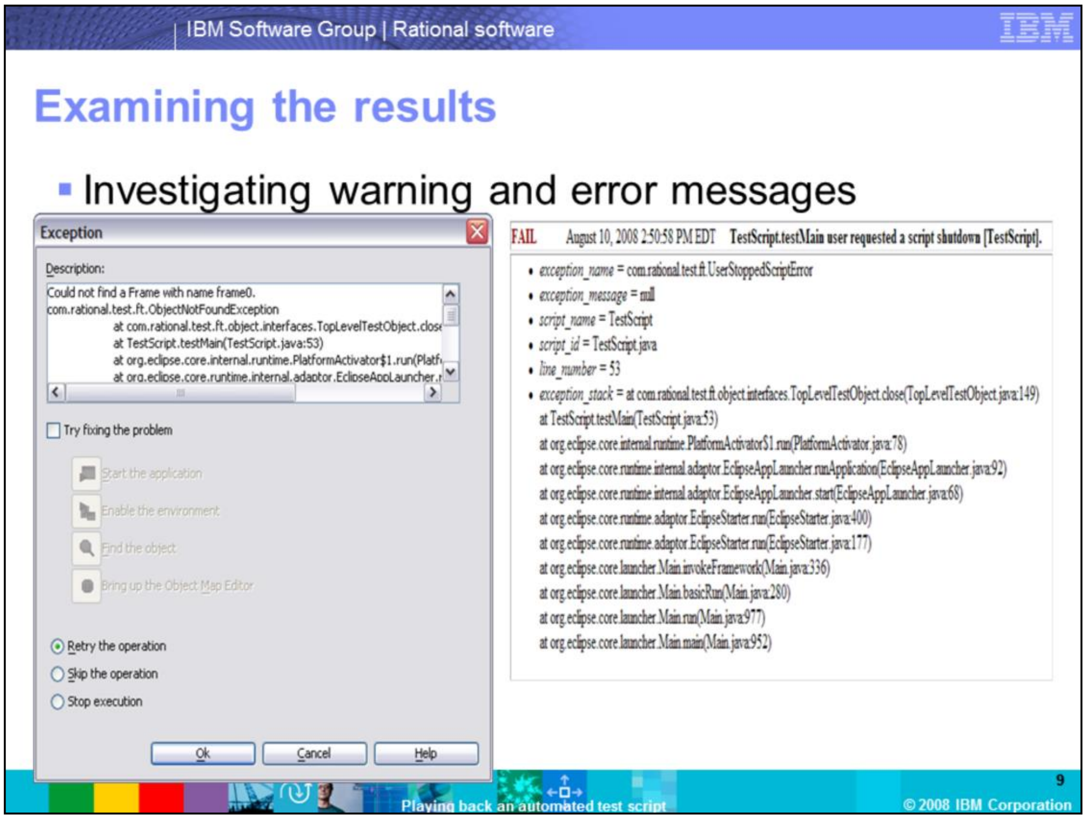
View Results

Applet ComparatorApplet started

Playing back an automated test script 8 © 2008 IBM Corporation

After playback is complete, the results display in an HTML log in a browser window. HTML is the default logging option. If the tester chose a different format for logging, the log displays in that format. This log is also found in the log folder in the Functional Test Projects view.

The left frame of the log provides quick navigation to important information. The right frame provides details regarding which parts of the script passed or failed.



Examining the results

- Investigating warning and error messages

```
Could not find a Frame with name frame0.
com.rational.test.ft.ObjectNotFoundException
        at com.rational.test.ft.object.interfaces.TopLevelTestObject.close
        at TestScript.testMain(TestScript.java:53)
        at org.eclipse.core.internal.runtime.PlatformActivator$1.run(Platf
        at org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.r
```

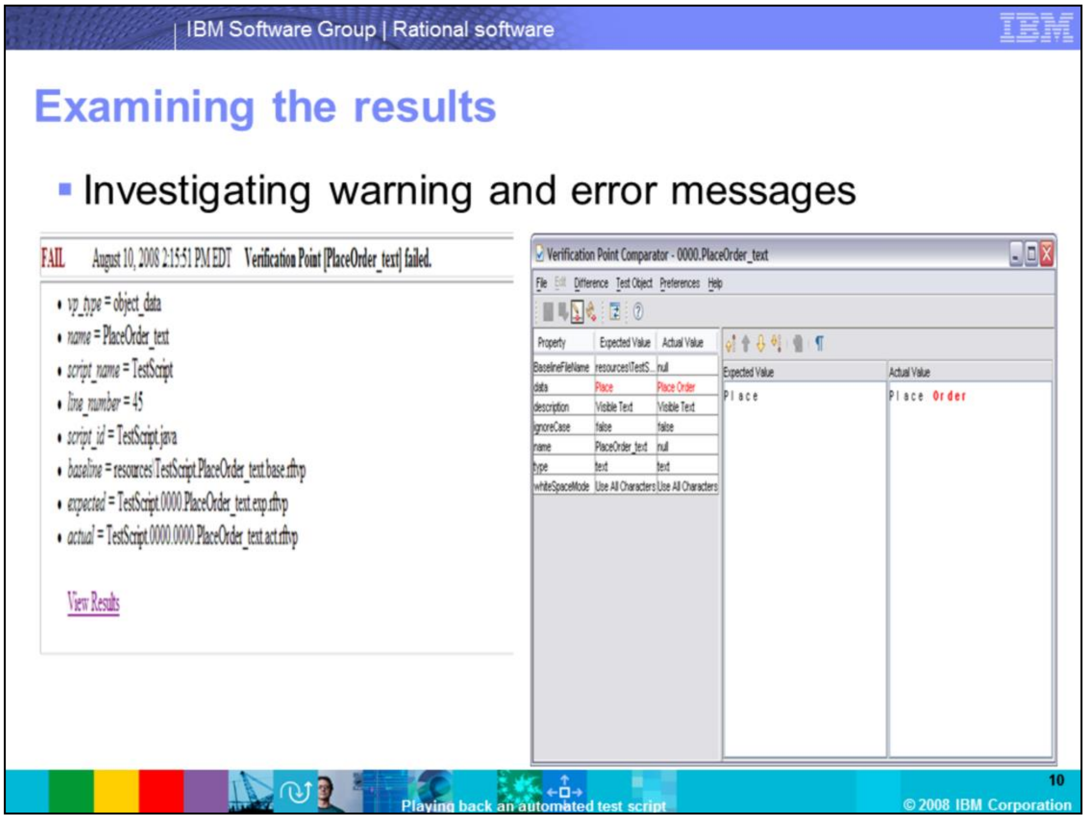
[ ] Try fixing the problem

[✓ Retry the operation] Skip the operation Stop execution

```
FAIL    August 10, 2008 2:50:58 PM EDT    TestScript.testMain user requested a script shutdown [TestScript].

  • exception_name = com.rational.test.ft.UserStoppedScriptError
  • exception_message = null
  • script_name = TestScript
  • script_id = TestScript.java
  • line_number = 53
  • exception_stack = at com.rational.test.ft.object.interfaces.TopLevelTestObject.close(TopLevelTestObject.java:149)
    at TestScript.testMain(TestScript.java:53)
    at org.eclipse.core.internal.runtime.PlatformActivator$1.run(PlatformActivator.java:78)
    at org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.runApplication(EclipseAppLauncher.java:92)
    at org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.start(EclipseAppLauncher.java:68)
    at org.eclipse.core.runtime.adaptor.EclipseStarter.run(EclipseStarter.java:400)
    at org.eclipse.core.runtime.adaptor.EclipseStarter.run(EclipseStarter.java:177)
    at org.eclipse.core.launcher.Main.invokeFramework(Main.java:336)
    at org.eclipse.core.launcher.Main.basicRun(Main.java:280)
    at org.eclipse.core.launcher.Main.run(Main.java:977)
    at org.eclipse.core.launcher.Main.main(Main.java:952)
```

Typical errors or warning messages in Functional Tester include “Object Not Found” and “Object Recognition is weak.” Review the error details in the application log to determine if an object is missing from the previous version on which the script was recorded. This should indicate why playback produces one of these error messages. Take the necessary steps to correct these errors or warnings. This ranges from removing the missing objects from the object map and script to updating the recognition properties of the object in question.



IBM Software Group | Rational software

Examining the results

- Investigating warning and error messages

FAIL August 10, 2008 2:15:51 PM EDT Verification Point [PlaceOrder_text] failed.

- *vp_type* = object_data
- *name* = PlaceOrder_text
- *script_name* = TestScript
- *line_number* = 45
- *script_id* = TestScript.java
- *baseline* = resources\TestScript.PlaceOrder_text.base.rftvp
- *expected* = TestScript.0000.PlaceOrder_text.exp.rftvp
- *actual* = TestScript.0000.0000.PlaceOrder_text.act.rftvp

View Results

Verification Point Comparator - 0000.PlaceOrder_text

Property	Expected Value	Actual Value
BaselineFileName	resources\TestS...	null
data	Place	Place Order
description	Visible Text	Visible Text
ignoreCase	false	false
name	PlaceOrder_text	null
type	text	text
whiteSpaceMode	Use All Characters	Use All Characters

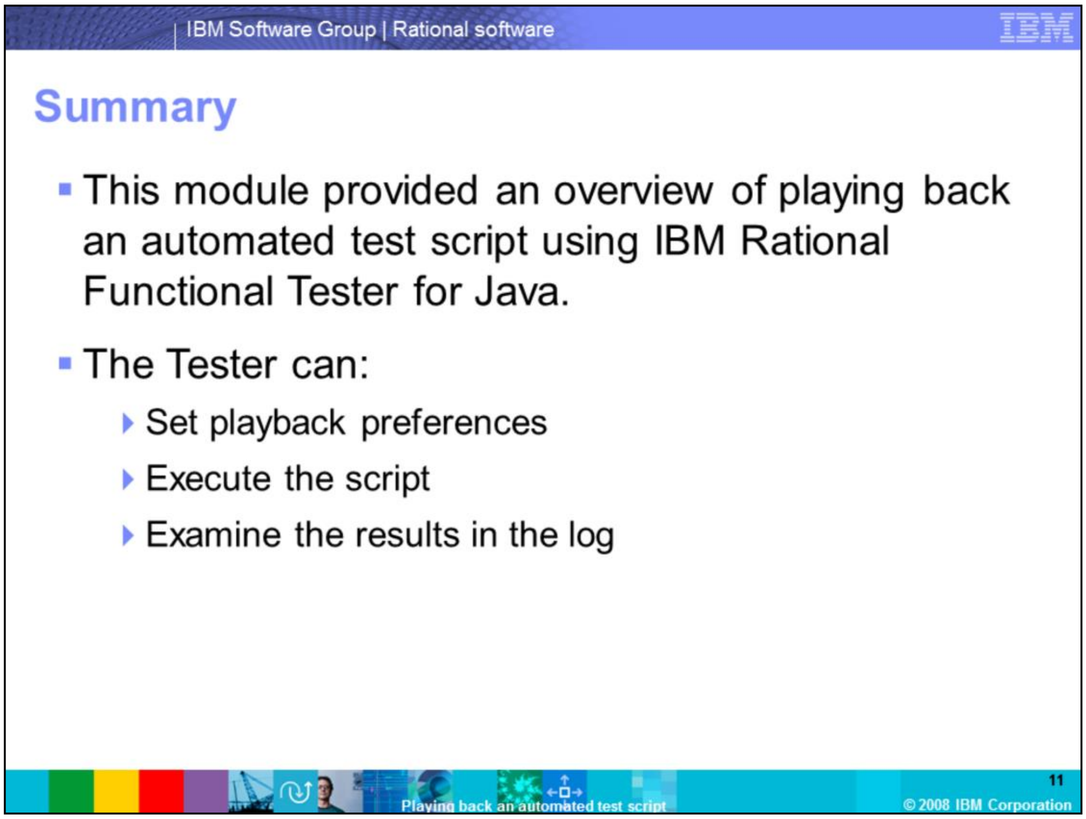
Expected Value	Actual Value
Place	Place Order

Playing back an automated test script

10

© 2008 IBM Corporation

Another part of the log is failed verification points. Find the failed verification point in the log and click the “View Results” link in that section. The verification point comparison window appears. This window shows the expected value recorded during the script recording and the actual value the script found during the playback. The difference in the value is highlighted in red.



IBM Software Group | Rational software

Summary

- This module provided an overview of playing back an automated test script using IBM Rational Functional Tester for Java.
- The Tester can:
  - Set playback preferences
  - Execute the script
  - Examine the results in the log

Playing back an automated test script

11

© 2008 IBM Corporation

This module provided an overview of playing back an automated test script using IBM Rational Functional Tester for Java. In this process, the tester sets playback preferences accordingly, executes the script, and examines the results in the log, searching for any problems.

Trademarks, copyrights, and disclaimers