



IBM Rational Software Conference 2009
As Real as It Gets!



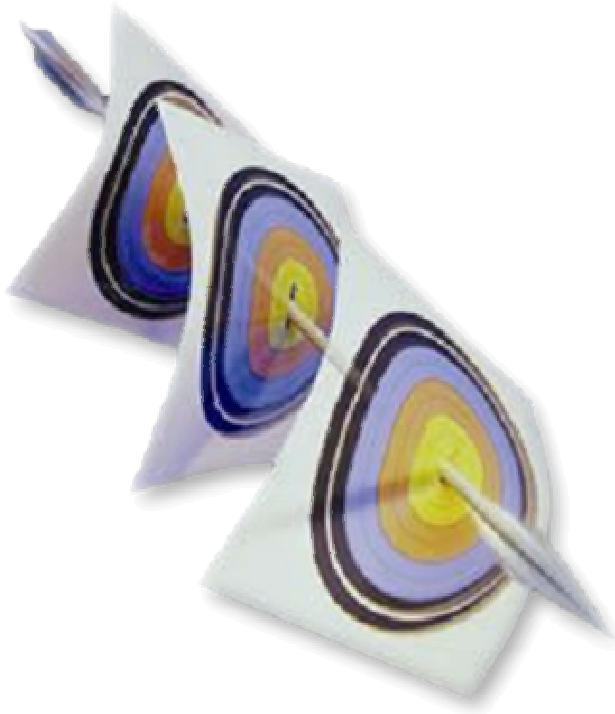
IBM Rational Software Conference 2009

Simplified Scripting – The new face of Rational Functional Tester (RFT)

SHINOJ ZACHARIAS
(shinoj.zacharias@in.ibm.com)

Rational software

Agenda



- Introduction to Rational Functional Tester
- Introduction to Simplified Scripting feature
- Record a Simplified Script
- Understanding Functional Tester simplified scripts.
- Editing of simplified script
- Application view and Application visuals
- Properties view
- Script execution and playback log
- Enabling the simplified scripting feature in IBM Rational Functional Tester
- Advanced features
- Demo
- Q & A

Introduction – Rational Functional Tester (RFT)



- Automated regression testing tool that lets users test Java, .NET, Windows, terminal-based, SAP, Siebel, Flex/Flash and Web applications running on Windows® and Linux platforms.
- Minimize test maintenance with scripts resilient to application changes
- Wizard enhanced automation to speed test creation for the new user
- Powerful scripting language and IDE for the professional tester
- **RFT Recorder**
 - Test scripts are recorded on the fly, as user navigates application
 - Verification points are inserted to validate system response
- **RFT Scripts**
 - Java code or VB.net is added to perform a variety of functions
 - Typical Modifications: Conditional branching, datapooling, refactoring
- **RFT Playback**
 - Scripts are executed
 - Discrepancies are logged



Functional Test - Project1/Script2.java - Rational Functional Tester

File Edit Source Refactor Navigate Search Project Script Configure Run Window Help

Script2.java

```

    */
    public void testMain(Object[] args)
    {
        startApp("ClassicsJavaA");

        // Frame: ClassicsCD
        tree2().click(atPath("Composers->Schubert->Location(PLUS_MINUS)"));
        tree2().click(atPath("Composers->Schubert->Symphonies Nos. 5 & 9"));
        placeOrder().click();

        // Frame: Member Logon
        ok().click();

        // Frame: Place an Order
        cardNumberIncludeTheSpaces().click(atPoint(31,10));
        placeAnOrder().inputChars("123456");
        expirationDate().click(atPoint(13,10));
        placeAnOrder().inputChars("12/12/2009");
        creditCombo().click();
        creditCombo().click(atText("Amex"));
        placeOrder2().click();

        //
        yourOrderHasBeenReceivedYourOr().performTest(YourOrderHasBeenR
        ok2().click();

        // Frame: ClassicsCD
        classicsCD(ANY,MAY_EXIT).close();
    }
    
```

Script2

- Script2Helper
- RationalTestScript
- Test Datapool
- Verification Points
- YourOrderHasBeenReceivedYourOr_text
- Test Objects
 - Private Test Object Map
 - cardNumberIncludeTheSpaces
 - classicsCD
 - creditCombo
 - expirationDate

Private Test Object Map for Script Script2

File Edit Find Test Object Preferences Applications Display Help

- Java: Frame: orderForm: javax.swing.JFrame
 - Java: Button: placeOrder: javax.swing.JButton
 - Java: ComboBox: creditCombo: javax.swing.JComboBox
 - Java: Text: cardNumberField: javax.swing.JTextField
 - Java: Text: expireField: javax.swing.JTextField
 - Java: Button: ok-orderlogon: javax.swing.JButton
- Java: Frame: logFrame1: javax.swing.JFrame
- Java: Frame: ClassicsJava: ClassicsJava
 - Owned: Java: Dialog: Message: javax.swing.JDialog
 - Java: Panel: JOptionPane: javax.swing.JOptionPane
- Java: Frame: ClassicsCD: ClassicsJava
 - Java: Button: placeOrderButton2: javax.swing.JButton
 - Java: Tree: tree2: javax.swing.JTree

Property	Value	Weight
class	javax.swing.JButton	100
classIndex	0	50
accessibleContext.accessibleName	OK	100
name	ok-orderlogon	100

Properties Tasks Console Problems

Pr... Value

Writable Smart Insert 22 : 5

Scripts in java

Object map



Introduction – Simplified Scripting



Record

Enhance

Execute

- Major usability improvement feature introduced in RFT version. 8.1
- Visual approach to automated test creation
- Natural language description of test actions
- Simplified Test Authoring
- Enhancing scripts without having the application to be running.
- Visuals (Screen Snapshots) showing objects interacted during recording.
- Storyboard navigation of test flow based on application screenshots.
- No knowledge of programming language such as Java is required.



Record Simplified Script

- Records
 - User interactions
 - Snapshots/Visuals
 - Object information
 - Verification point data
 - Data drive
- Shows minimized recorder toolbar
- Natural language statements in Recorder Monitor
- Recorder window status
 - Recording
 - Paused

Paused

Getting Started

To Record a script:

1. Select to configure and start the application under test
2. Record the script
3. Select to stop recording

You can also...

Select to pause recording

Select to add verification points to your script

Select to data-drive your script

Select to turn off Getting Started help

```
Recording Script Script2 Started
Recording Script Script2 Paused
Start Application ClassicsJavaA
Recording Script Script2 Resumed
Click Place Order
Click OK
Click Card Number (include the spaces)
Type 123456789
Click Expiration Date
Type 12/12/2009
Click Place Order
Click OK
Recording Script Script2 Paused
```

Paused

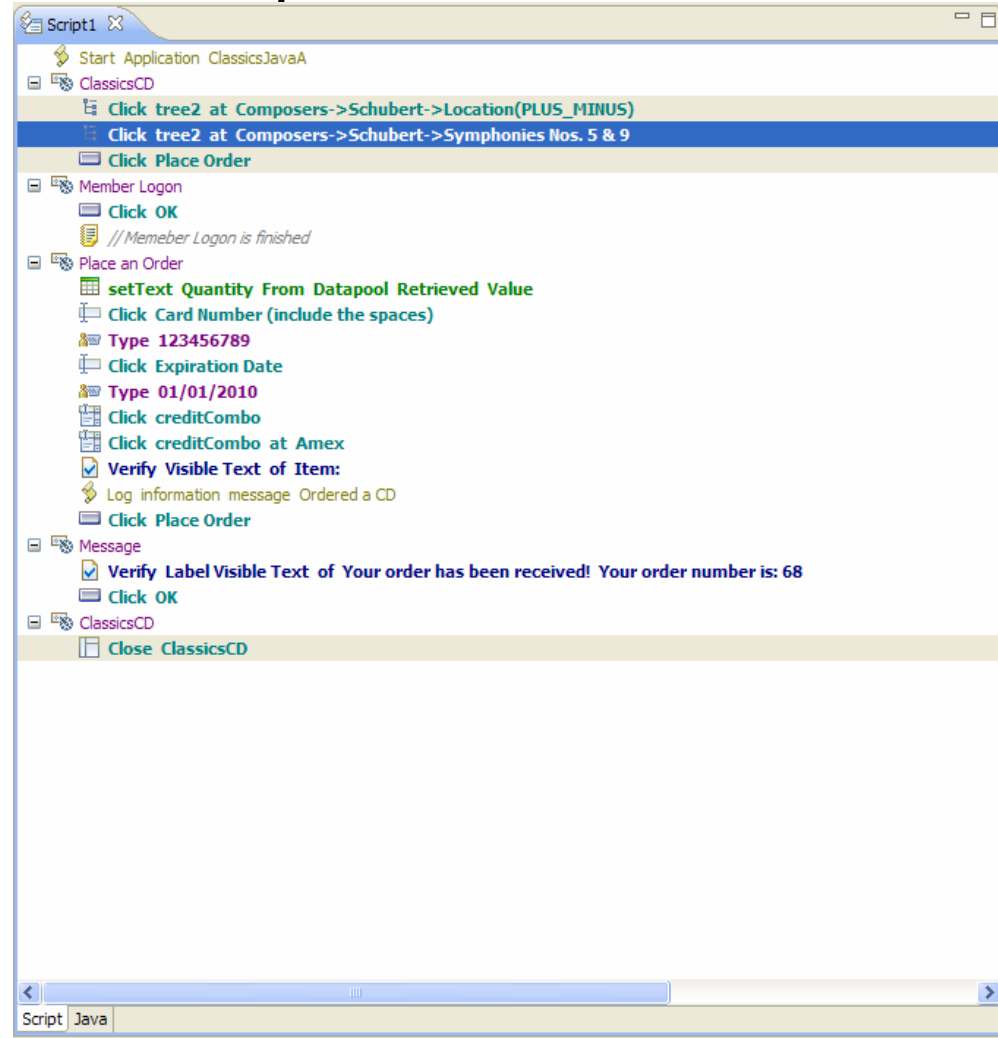
The screenshot displays the IBM Rational Functional Tester interface. The main window is titled "Functional Test - DemoProject/Script1.rftss - Rational Functional Tester". The interface is divided into several panes:

- Script Explorer:** Shows a tree view of the test script. A callout labeled "Simplified script" points to the script content, which includes actions like "Click tree2 at Composers->Schubert->Location(PLUS)", "Click tree2 at Composers->Schubert->Symphonies Nos. 5 & 9", "Click Place Order", "Member Logon", "Click OK", "Message", and "Verify Label Visible Text of Your order has been received! Your order number is: 68".
- Application View:** Displays a screenshot of the application being tested, titled "ClassicsCD". It shows a menu with "Composers" (Schubert, Haydn, Bach, Beethoven, Mozart) and an album page for "Schubert" (String Quartets Nos. 4 & 14) with a "Place Order" button. A callout labeled "Application visuals" points to this view.
- Script Tab:** A callout labeled "Script tab" points to the "Script1" tab in the "Functional Test Projects" pane.
- Java Tab:** A callout labeled "Java tab" points to the "Script Java" tab in the "Properties" pane.
- Properties View:** A callout labeled "Properties view" points to the "Properties" pane, which shows details for the "tree2" control, including its name, action ("Click Tree Node"), and path ("Composers->Schubert->Symphonies Nos. 5 & 9").
- Thumbnails:** A callout labeled "Thumbnails" points to a row of five small thumbnail images at the bottom of the interface, representing different stages of the test execution.



Understanding RFT simplified scripts.

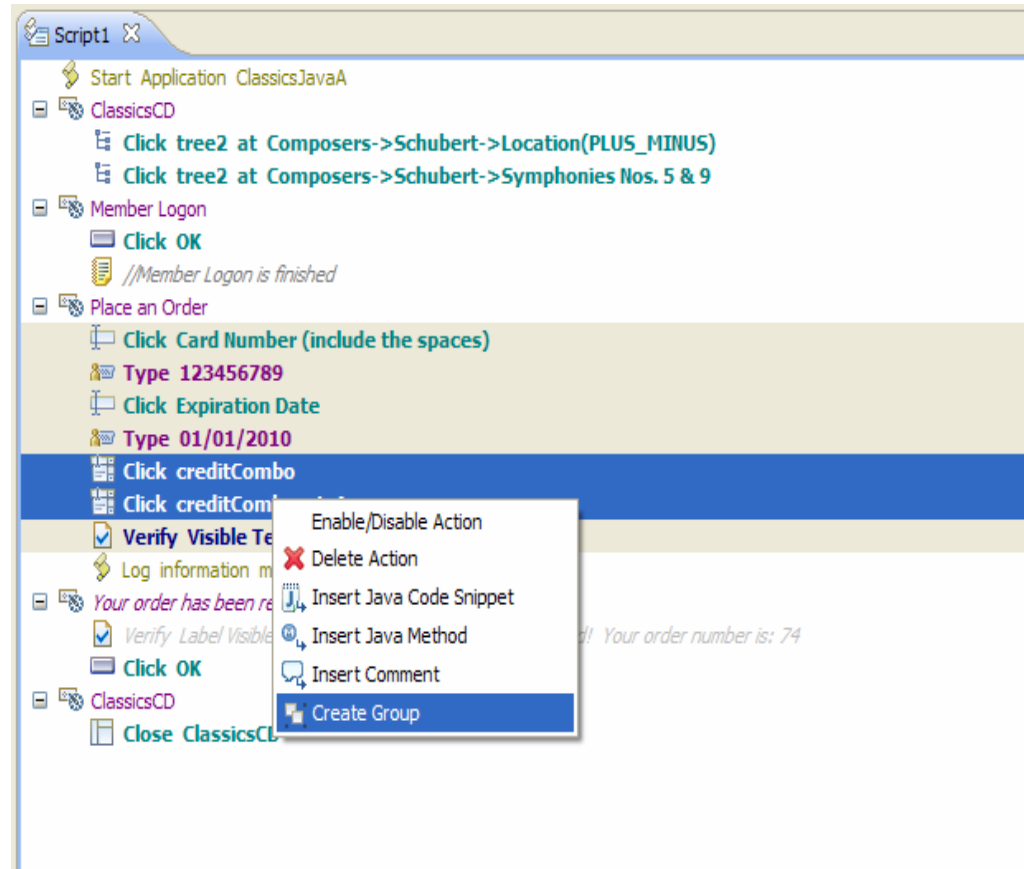
- Simple, easy to understand English like statements.
- Grouping with respect to pages
- Easy visualization of objects/groups in Application view
- Operations on script line/group
- Drag and drop any where
 - Statements and groups
- Statements are color coded
 - Verification point
 - Data drive
 - Script utility methods
 - Normal script statements



Enhancing Script

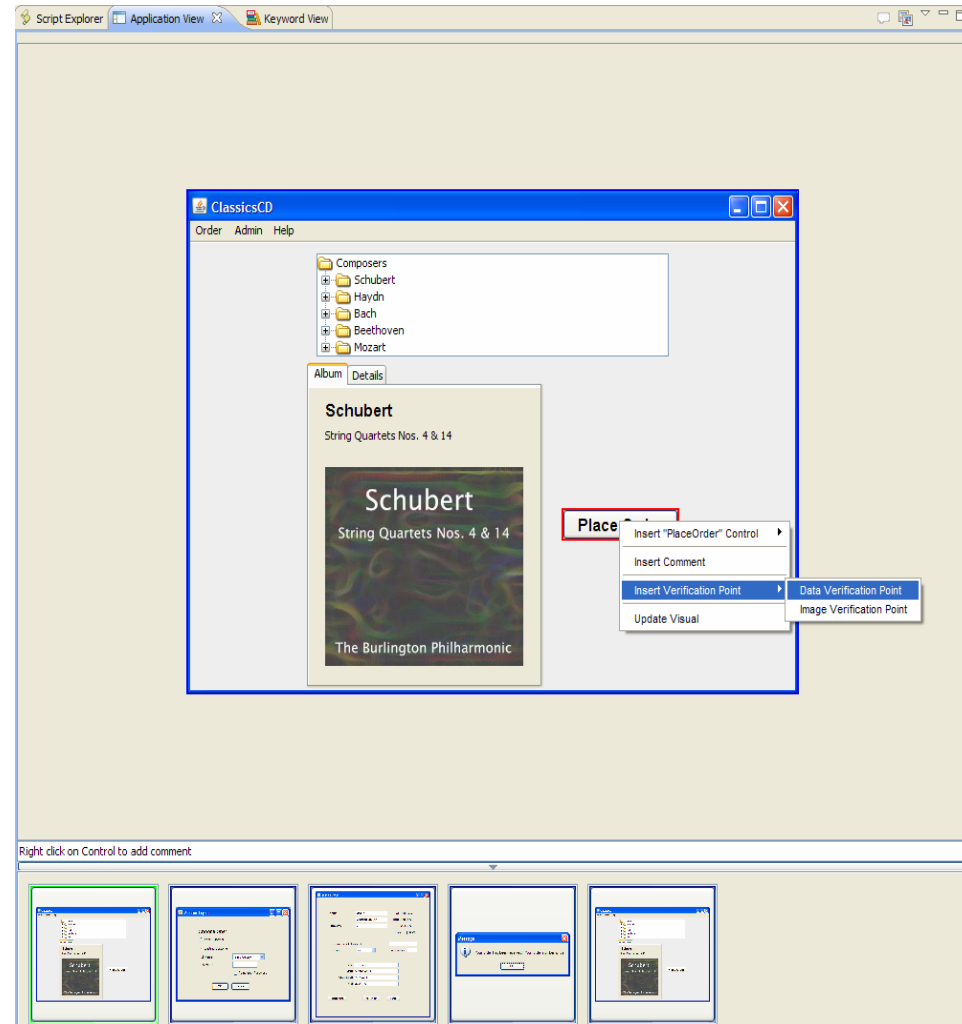


- Easy modification
- No knowledge of program language required
- Each script line/group has a contextual menu
 - Enable/Disable
 - Deleting script
 - Create group
- Inserting comments
- Inline editing
- Drag and drop
- Custom code/module
 - Java Code Snippet
 - Insert Java Method



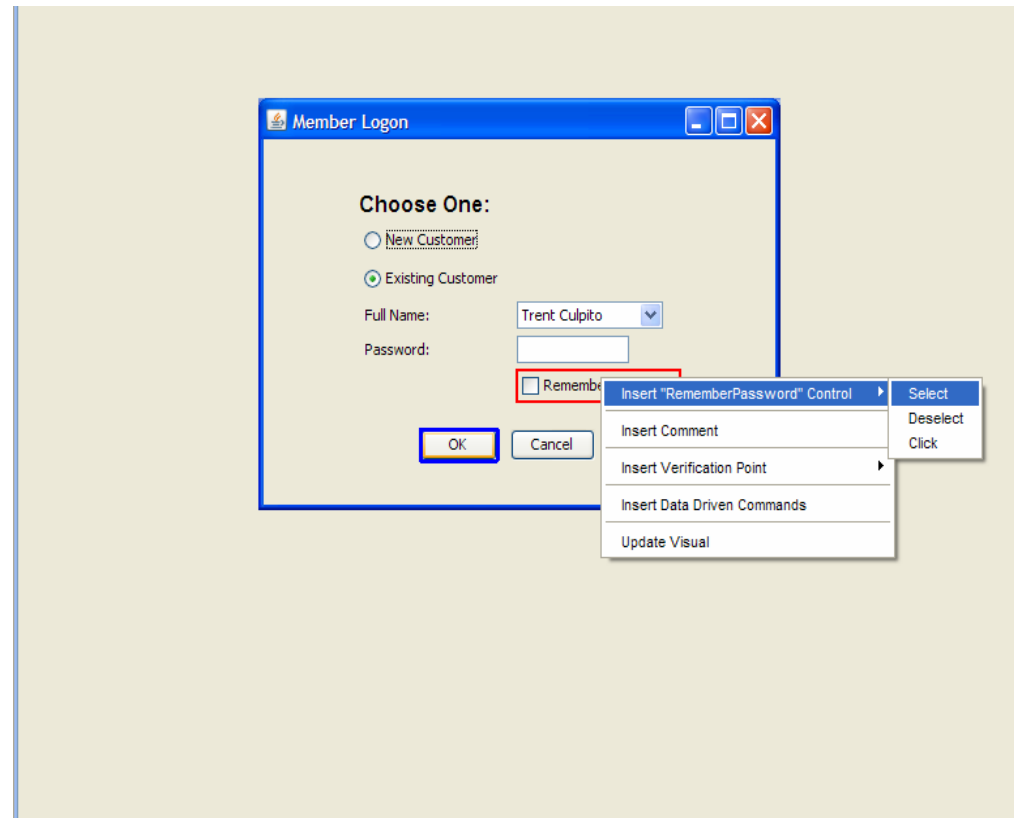
Application view and Application Visuals

- Snapshots of windows/pages
- Persists object's properties, data vp and data drive information.
- Object highlighting
 - Blue, interacted object during recording
 - Red, object that can be automated
- Storyboard testing
 - pages in the order in which the it was visited during recording.
 - Options to show the thumbnail image of the currently active script or all the visuals of the project. The option provided in the Application view toolbar
 - Create a test 'a story' from the visuals.



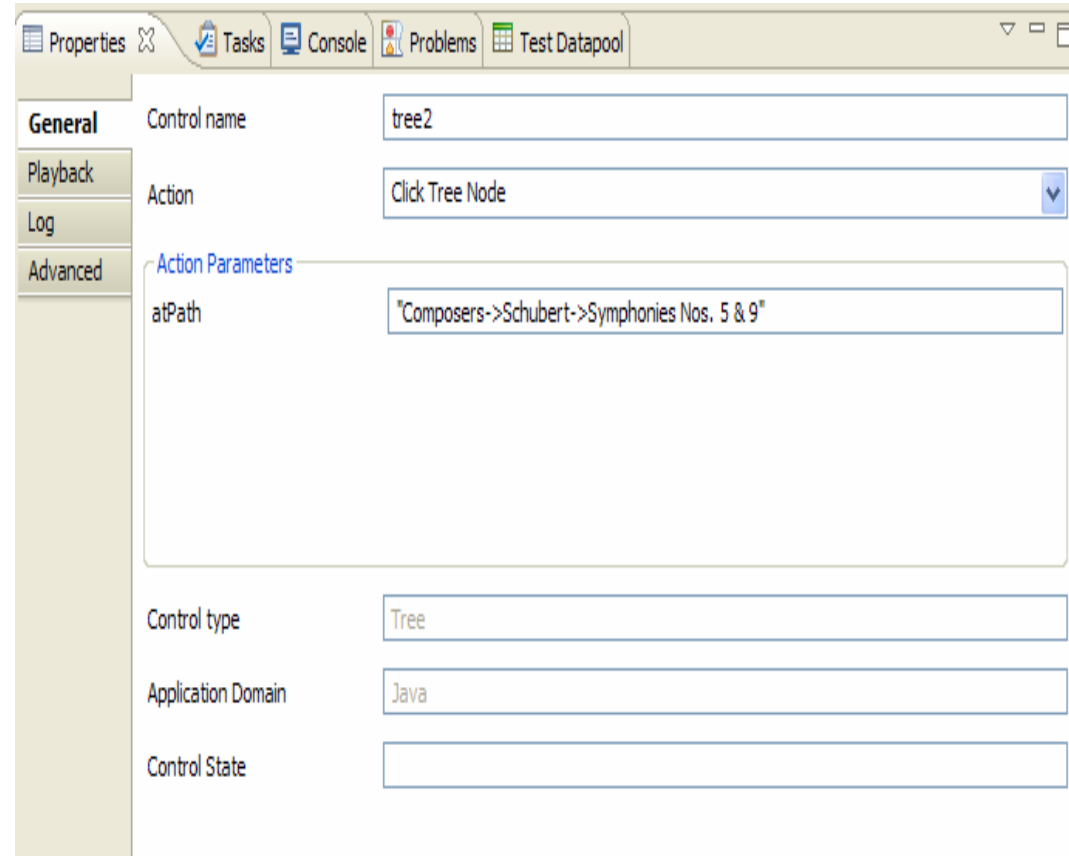
Application view and Application Visuals – Continued.

- Simplifies test script modification
- Right clicking on the object provides a list of actions that can be performed on the object such as:
 - Inserting control with specific action
 - Inserting comments
 - Inserting Data and Image verification
 - Group verification points
 - Inserting Data Drive Commands
 - Updating Visuals



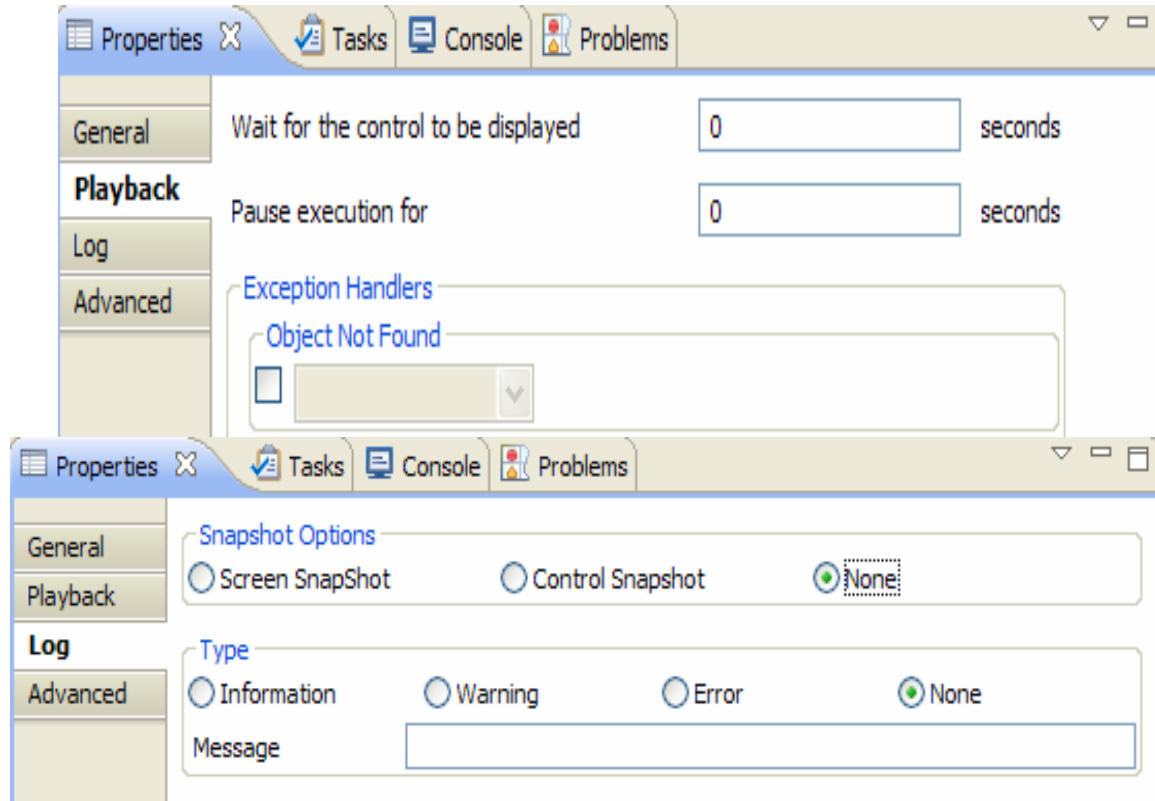
Properties View

- Properties view displays the details about the selected simplified script line. Properties view consists of four tabs
 - General
 - Playback
 - Log
 - Advanced
- Script is editable from the properties view.
- The General tab
 - Control information.
 - Action and action parameter



Properties View – Continued

- The Playback tab
 - Exception handling
 - Wait for the object
 - Delay Execution
- The Log tab
 - Log some information
 - Object/Screen snapshot
- The Advanced tab.
 - Verification point data
 - Opens vp in Verification Point Editor



Script Execution and Playback log



- Executes the simplified script
- Playback Monitor shows simplified script statements.
- The playback log shows verdict of the run as well as the simplified script statement.
- Simplified Script Line Number shows which line in the simplified script editor the verdict belongs to.

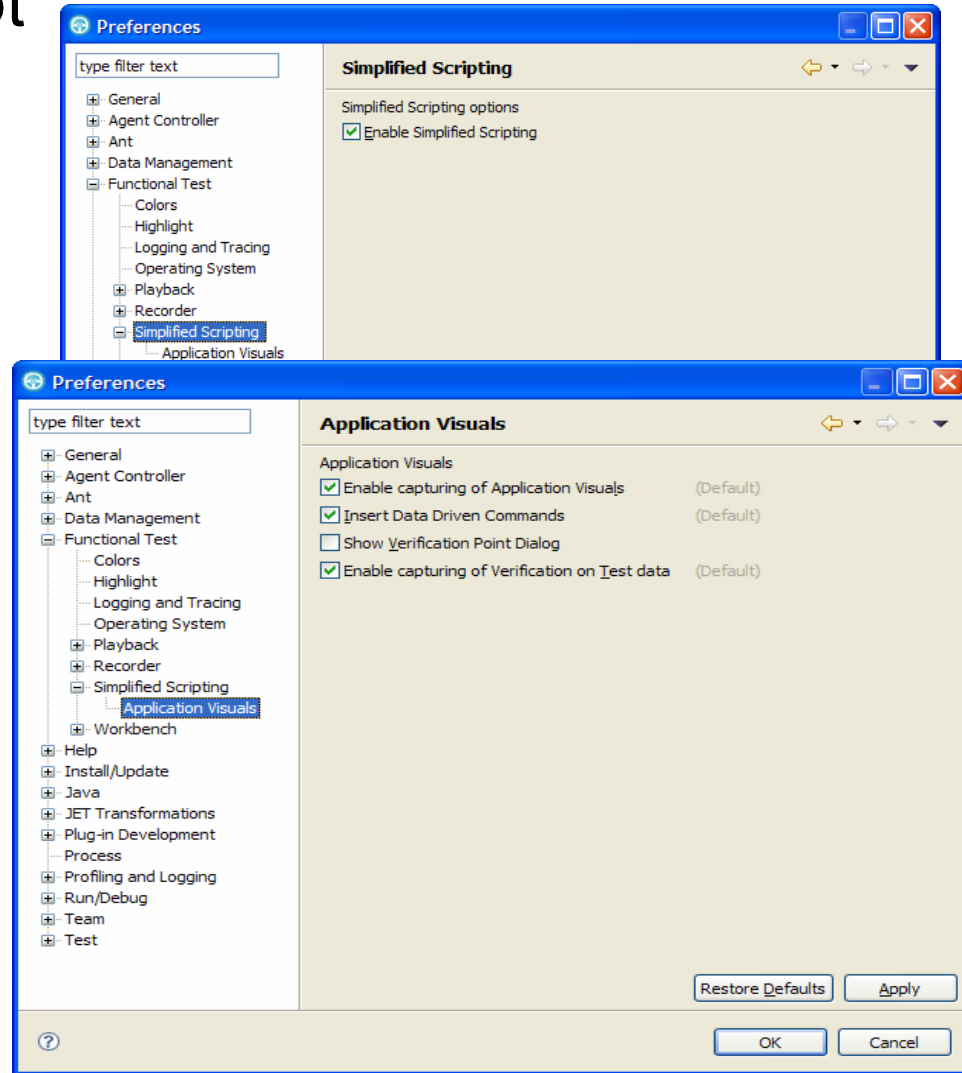
The screenshot displays the Rational Functional Tester (RFT) interface. At the top, a 'Playback' window shows 'Script1' with a 'Click OK' action. Below it, the 'Mozilla Firefox' browser window is visible, displaying the Rational Functional Tester application. The main interface is divided into several sections:

- Test Verdict:** A section on the left with filters for Success's, Failures's, Warnings, and Messages. A tree view shows the script structure, including 'Script start [Script1]', 'Start application [Classics.JavaA]', and 'Click tree2 at Composers-> Schubert-> Location(PLUS_MINUS)'. Verification points are listed below, such as 'Verify Visible Text of Street' and 'Verify Visible Text of Item:'. A 'Failures Verdicts' and 'Warning Verdicts' section is at the bottom left.
- Overall Verdict:** A central area featuring a pie chart showing the overall results: 75.0% Success, 12.5% Warnings, 8.3% Failures, and 4.2% Messages. A summary table indicates: Success's: 18, Failures's: 2, Warnings: 1, Messages: 3. It also shows 'Successful Verification Points: 2' and 'Failed Verification Points: 1'.
- Log Entries:** A detailed log on the right shows the execution of individual script steps:
 - Script start [Script1]:** INFORMATION, 31-Jul-2009 05:57:48.284 PM. Details: Line Number = 1, Iteration Count = 0, Script Name = Script1, Script ID = Script1.java.
 - Start application [Classics.JavaA]:** PASS, 31-Jul-2009 05:57:48.316 PM. Details: Property Name = Classics.JavaA, SimplifiedScript Line Number = 1, Line Number = 34, Script Name = Script1, Script ID = Script1.java.
 - Click tree2 at Composers-> Schubert-> Location(PLUS_MINUS):** PASS, 31-Jul-2009 05:57:50.822 PM. Details: SimplifiedScript Line Number = 3, Line Number = 40, Script Name = Script1.
- Datapool Values:** A section at the bottom right for data collection.



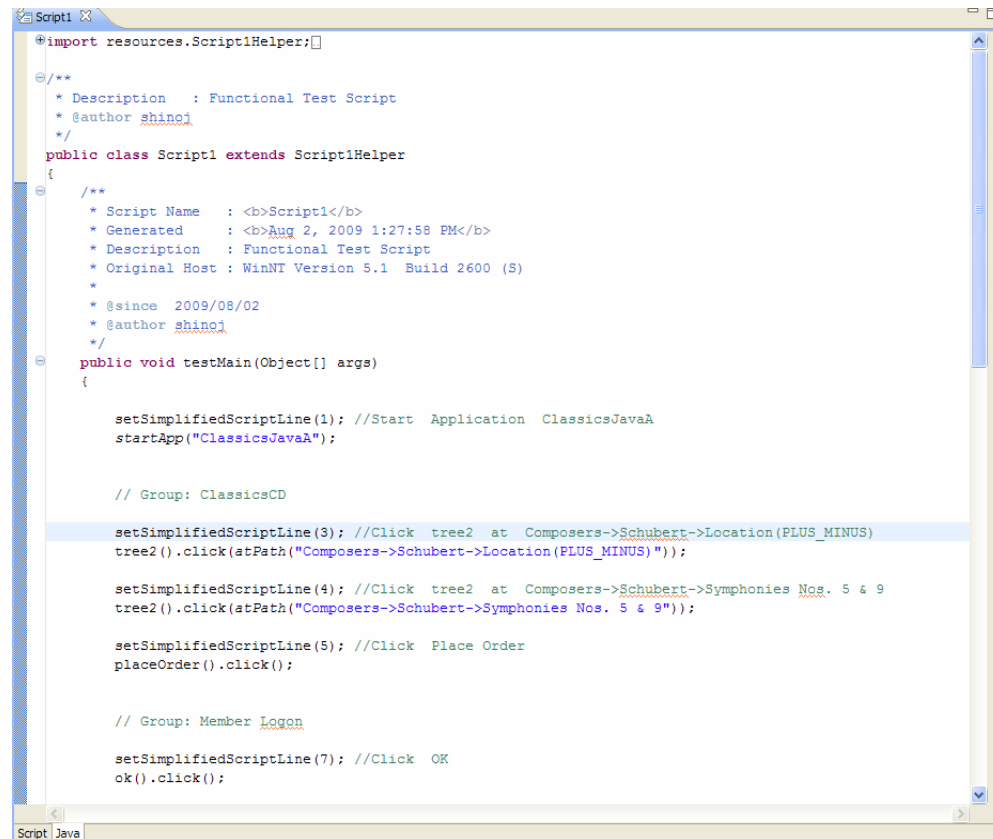
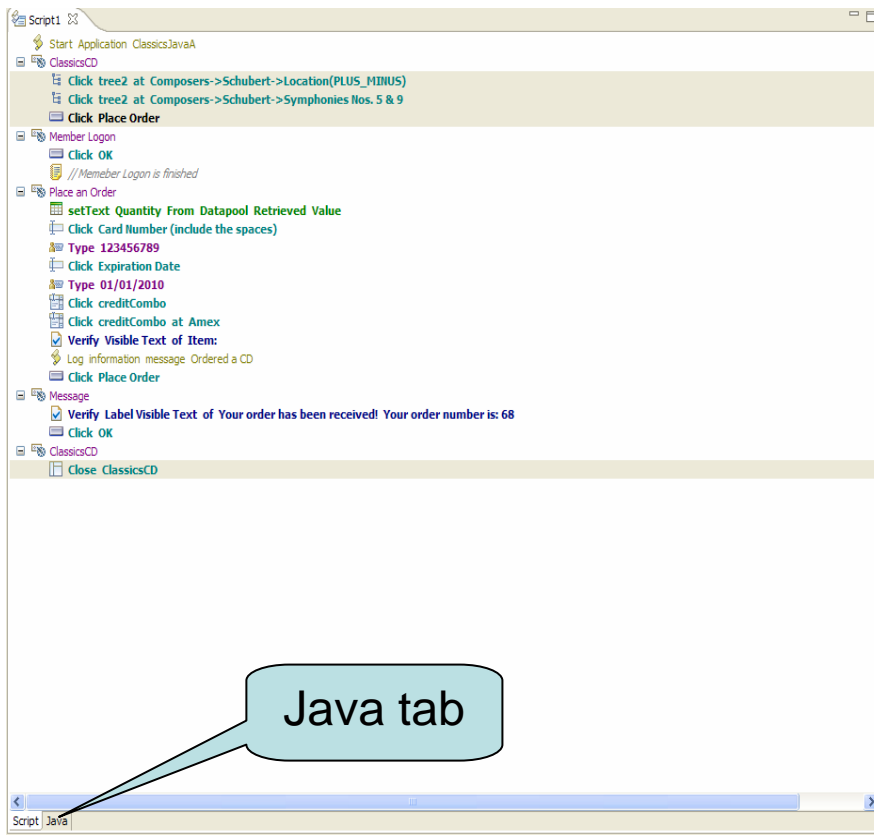
Enable the simplified script

- Enablement from RFT preference
- Enabled by default
- Simplified Scripting preference
 - [Enable Simplified Scripting](#)
- Application visuals preference
 - [Enable Capturing of Application Visuals](#)
 - [Insert Data Driven Commands](#)
 - [Show verification point dialog](#)
 - [Enable capturing of Verification on Test Data](#)



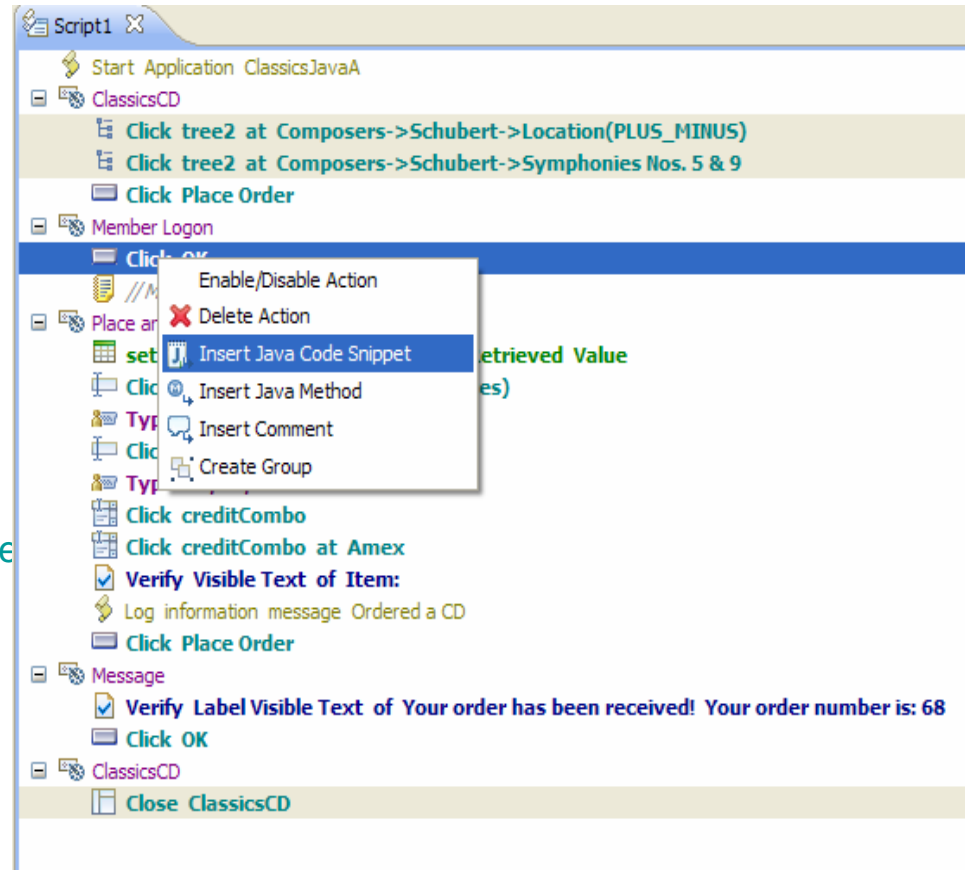
Advanced features

- When simplified script is run, RFT generates the java and execute the java. To view the generated java, click on the Java tab in the Simplified Script editor.



Advanced features – Java code snippet/Module

- Simplified Scripts have the file extension .rftss
- Java code gets regenerated
- Changes made directly to Java lost
- Insert Java Code Snippet
 - Inserts a custom code section in java after the java code for the selected simplified script
- Insert Java Module
 - Create a skeleton Java module section in the generated code
 - Java modules section will be appear end of the generated java.
- Custom code can be added in the //TODO section of the generated code section/java module section.



Advanced Feature - Java code snippet/Module

```
setSimplifiedScriptLine(14); //Type 01/01/2010  
placeAnOrder().inputChars("01/01/2010");
```

Java code
snippet

```
setSimplifiedScriptLine(15); //MyCustomCode  
// BEGIN java code snippet MyCustomCode  
    //TODO: Add your code here.  
// END java code snippet MyCustomCode
```

```
setSimplifiedScriptLine(16); //Click creditCombo  
creditCombo().click();
```

```
setSimplifiedScriptLine(17); //Click creditCombo at Amex  
creditCombo().click(atText("Amex"));
```

Java
Module

```
// BEGIN java method JavaModule1  
public static void JavaModule1()  
{  
    //TODO: Add your code here  
}  
// END java method JavaModule1
```



Demo – Simplified Scripting



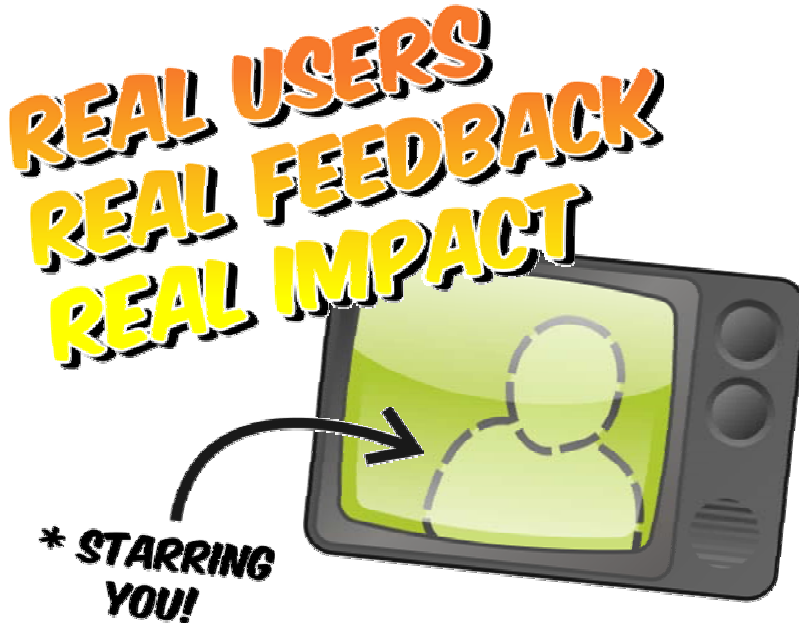


Thank You

© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.



IBM Rational User Technologies



You're invited to the **SVT Customer Quality Initiatives and Scenario Inspection Booth** on **13th August 2009**, to know more about CQI Programs, where you will get hands-on experience with the latest Rational products and speak one-on-one with technical personnel on topics including: usage scenarios, user interface design, ease-of-use, user assistance, learning, and quality.

This is a chance to share your reality with us through direct and specific feedback!

Sign up [SVT CQI Programs](#)
driven by ISL
Contact – Rahul Jaini
Email – rahuljaini@in.ibm.com

