

Automating Your OFBiz Testing Using Selenium

Brett G. Palmer
Automation Groups, Inc.



Agenda

- Why testing is needed for OS Projects like OFBiz
- Review of Selenium
- Automating Test Builds with Selenium Grid
- Introduction to Selenium Xml
- Other Test Areas



Maintaining OS Projects

- OS provides opportunities but not for free
- How do you maintain changing code?
 - One time snapshot
 - Maintain own repository
 - Test continuously



Snapshot Approach

- Good:
 - Saves you a lot of startup costs
- Bad:
 - You now have to maintain it
 - Little help from community



Own Repository

- Good:
 - More control of your source code
 - Ability to change and merge project code
- Bad:
 - Requires discipline to update frequently
 - Merging can be difficult
 - Often go months/years without merging



Continuous Testing

- Requires good test coverage for your application
- Stay close to the trunk
 - Developers update frequently
 - Production lags by a month



Continuous Testing

- Good:
- Constantly updated by project
 - Lowers maintenance cost
 - Benefits the community
- Bad:
 - Adds risk to production (depends on testing)
 - Manual production DB updates



Cost vs. Benefit for Testing

- Unit Testing
 - Less effective for mature platforms
- Functional Testing
 - Most defects found in application logic in business services
- System and Integration Testing
- Performance Testing
 - Some infrastructure defects found under load conditions



Automation Groups Test Tools

- Few Unit Tests
 - DOH/Rhino for Dojo/AJAX unit testing
- Selenium/Selenium XML
 - Functional and application testing
 - System Test
- Grinder
 - Service Testing
 - Performance and Load Testing



Selenium Introduction

- Apache 2.0 License
- Selenium tests run directly in the browser
- Doesn't emulate browser
 - Leverage features of browser rather than reinvent them (i.e. JavaScript)
- Supports multiple browsers
 - IE, Firefox, Safari, Opera, Others



Selenium Products

- Selenium Core
- Selenium IDE
 - Plugin for Firefox
- Selenium RC
 - Support Java, C#, Perl, Python, Ruby, and HTML
- Selenium Grid
 - Parallelize Selenium RC test environments

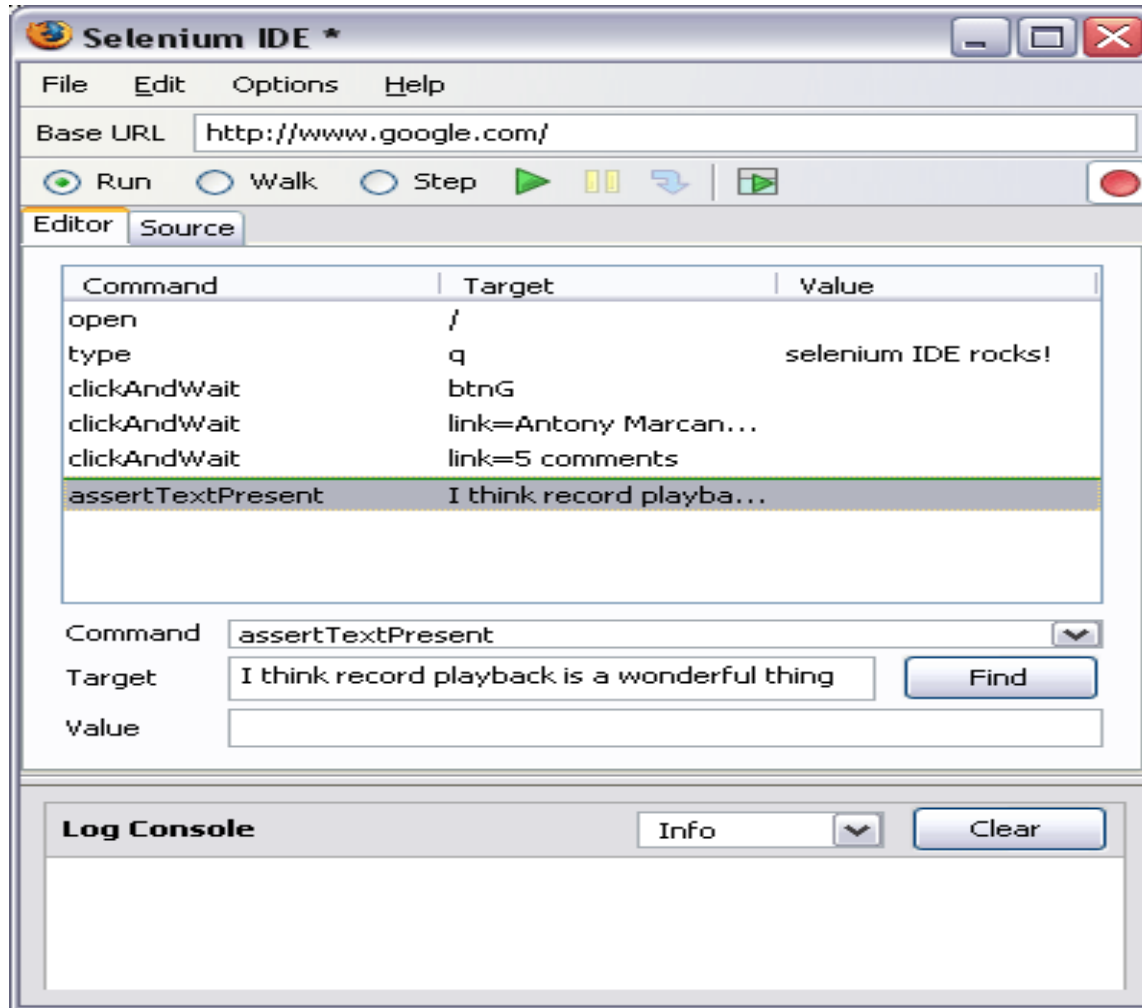


Selenium IDE

- Integrated development environment for Selenium
- Firefox Plugin
- Easily record, edit, and debug tests
- Intelligent field selection will use Ids, names, or XPATH as needed
- Save tests as HTML, Java, C#, Python, Perl, or Ruby



Selenium IDE

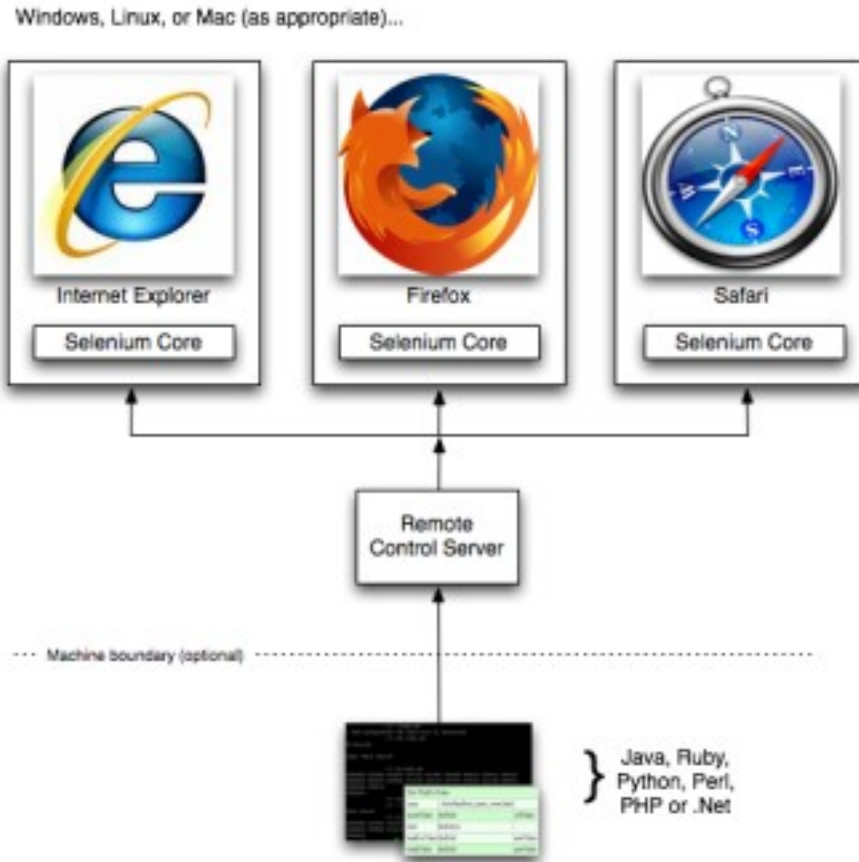


Selenium Remote Control (RC)

- Selenium RC comes in two parts
 - **Server**
 - Launches browsers
 - Acts as HTTP proxy for web requests
 - **Client Libraries**
 - Java, C#, Python, Ruby, Perl, HTML
 - Allows for test automation



Selenium RC



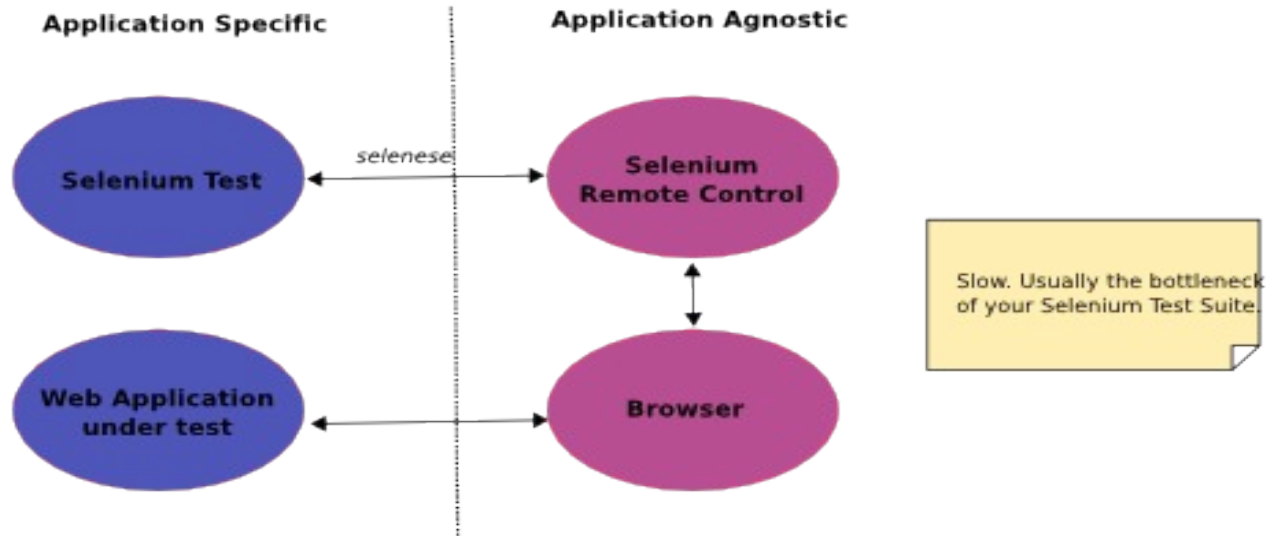
Selenium Grid

- Created to solve problems with RC
- Stability when running multiple browser tests
- Improve performance of web acceptance tests
- Easy to run against multiple environments



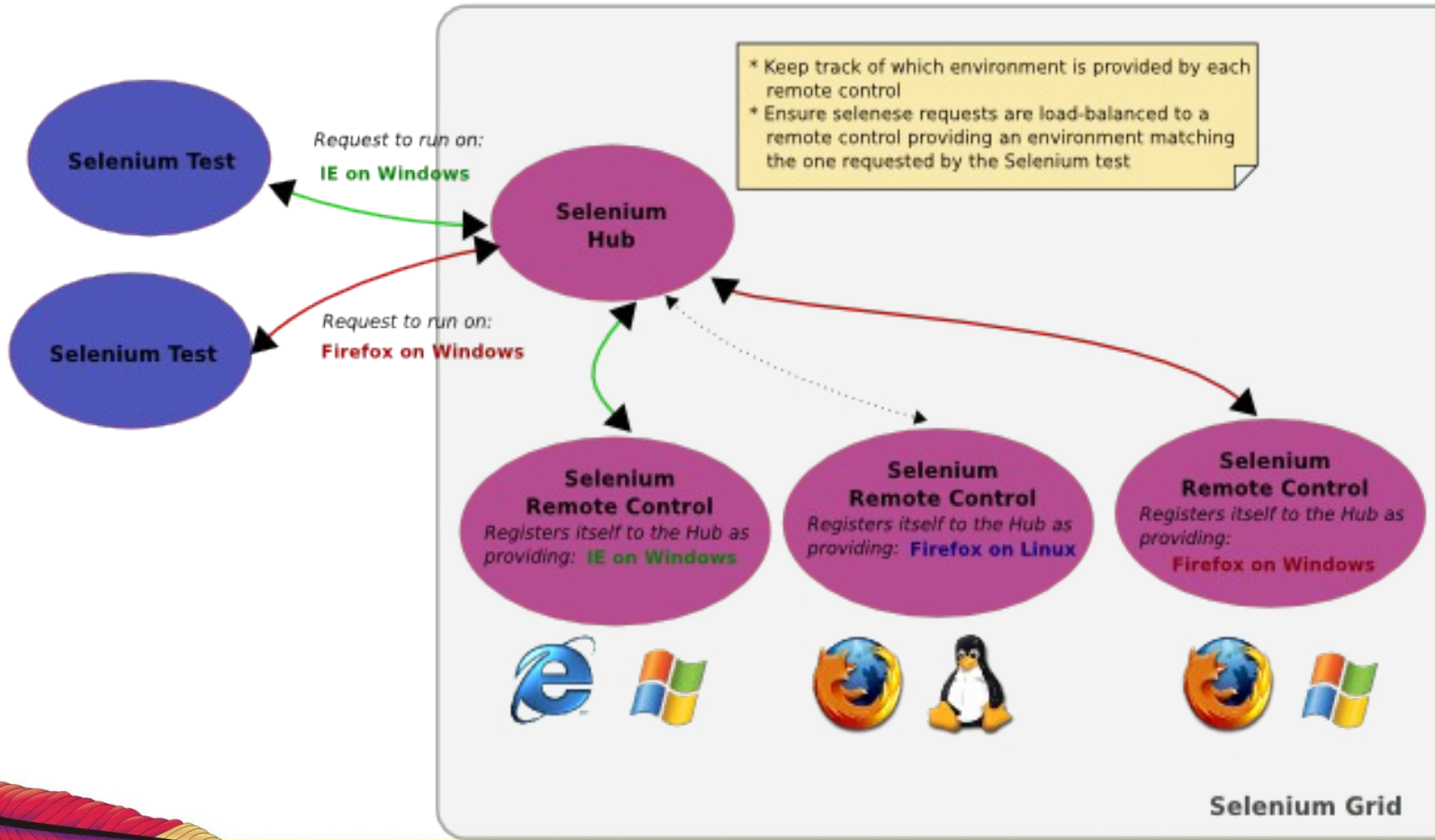
Selenium RC Setup

Traditional Selenium Setup

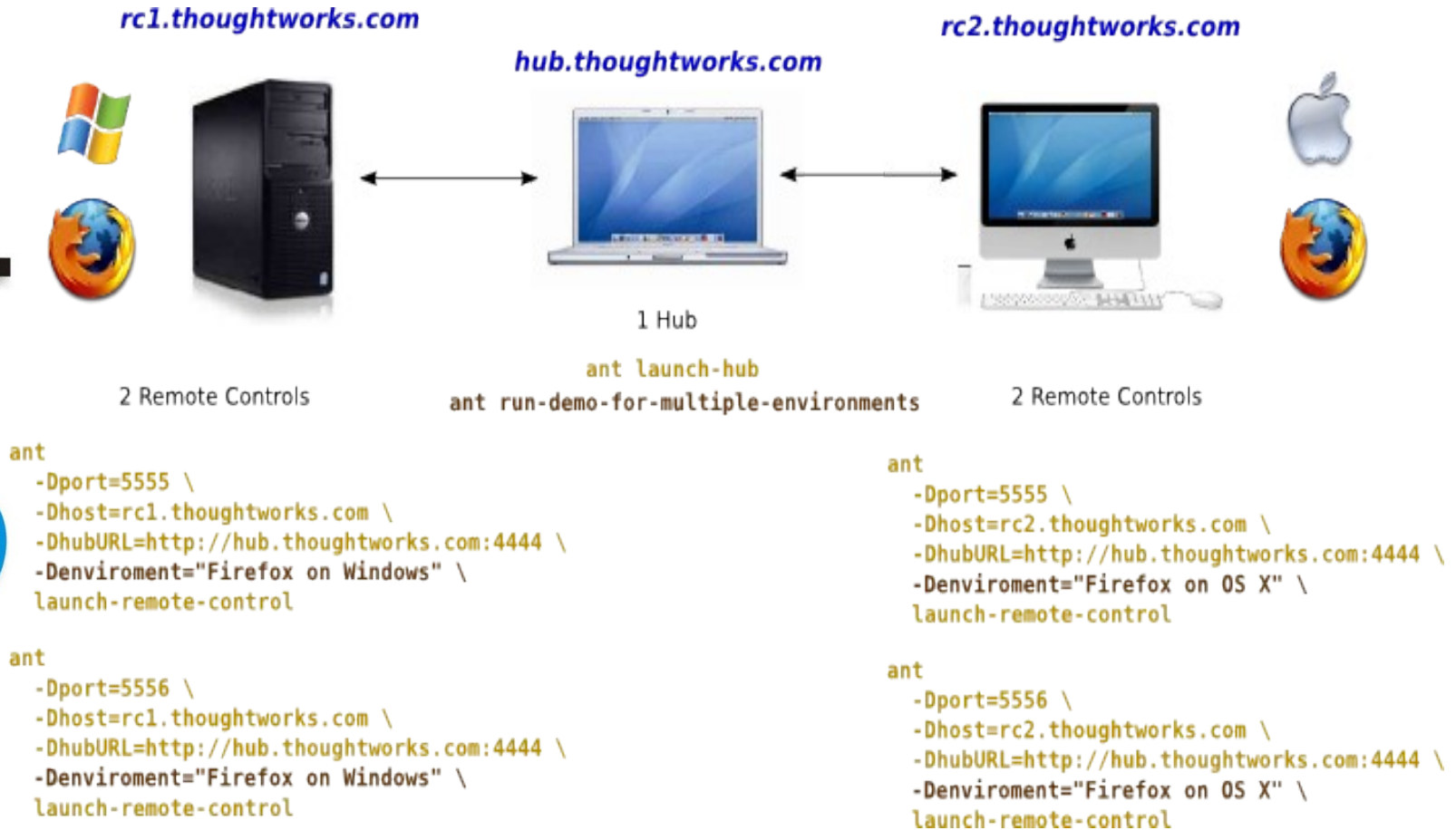


Selenium Grid Continued

Selenium Grid : Requesting a Specific Environment



Selenium Grid Example



Selenium XML Intro

- Selenium Java RC Extension
- Uses XML to create data driven tests
- Calls the Selenium RC APIs
- Added utility methods for capturing and storing data



Selenium XML Objectives

- Easy to use for testers and developers
- Separate test logic from data
- XML commands match RC APIs
- Run outside of OFBiz container
- Modular: Allow for suites of tests
- Extend: Ability to create plugins



Differences between XML and RC

- Selenium Java RC
 - Flexibility to write any test
 - Tests have to be compiled and run
 - Requires developers
- Selenium XML
 - Increase test coverage by modifying data, no compile necessary
 - Simple XML mark up for testers and developers
 - Jython scripts for custom plugins



Selenium XML Features

- Select HTML elements using
 - ID, name, CSS, DOM, Xpath
- Utilities for generating random data
- Capture Text by Regular Expression
 - `<captureText regExp="" group="" results="regResults" />`
 - `"xx <tag a=b> yy </tag> zz"`
 - `<captureText regex="<(\\S+?).*?>(.*?)<\\1>" group="2" results="xmlValue" />`



How To Create A Test

- Use Selenium IDE to record session
 - Easily captures fields and Ids
 - Does 70% of manual work
- Save testcase as HTML/XML
- Run Selenium to SeleniumXml converter
- Break into modules
 - E.g. functional area or page
- Parameterize data

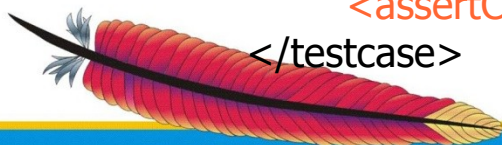


Sample Selenium Xml

```

<testcase>
<setup property="SampleSelenium"
    serverHost="127.0.0.1"
    serverPort="4444"
    browser="*firefox" />
    <uniqueId out="newId" />
    <uniqueId out="emailName" />
    <append src1="{emailName}" src2="@somewhere.com" out="email" />
    <randomString prefix="test_" size="25" out="newString" />
    <open value="http://localhost:8080/ecommerce/control/newcustomer" />
    <waitForPageToLoad value="3000" />
    <print value="Done waiting for page"/>
    <type name="USER_FIRST_NAME" value="Test_Sel_First" />
    <type name="CUSTOMER_CITY" value="Orem" />
    <type name="CUSTOMER_EMAIL" value="{email}" />
    <type name="USERNAME" value="{newId}" />
    <click locator="css=input[type='submit']"/>
    <waitForPageToLoad value="5000" />
    <getHtmlSource out="htmlSource"/>
    <assertContains src="{htmlSource}" test="performed successfully" />
</testcase>

```



Selenium Xml Testsuite

- Create Reusable Tests
- Combine Tests into Test Suites
- Data is shared between tests
- Data plugin
 - Uses Jython to read CSV
 - Loops through each row and applies data dynamically



Selenium Test Suite

```

<testcase>
  <setup property="SampleSelenium"
    serverHost="127.0.0.1"
    serverPort="4444"
    browser="*firefox" />
  <loadData file="userData.csv" iterations="-1">
    <testcase file="ofb_example_register.xml"/>
    <testcase file="ofb_example_login.xml"/>
    <testcase file="ofb_example_createOrder.xml"/>
  </loadData>
</testcase>
.....

```

Sample userData.csv file:

userLogin, passWord, firstName,lastName,address1,city,zip,state

user1, ofbiz, John,Walker,629 E. 1650 S.,Orem,84058,UT

user2, ofbiz, Walter,Walker,629 E. 1650 S.,Orem,84058,UT

user3, ofbiz, Bob,Walker,629 E. 1650 S.,Orem,84058,UT



Conclusion

- Recommended Process
 - Test Continuously
 - Update source often
- Recommended Test Tools
 - Selenium IDE to record Test Case
 - Use Selenium XML with Selenium Grid for acceptance and functional tests
- Other Test Tools
 - Grinder for performance testing



References

- Selenium IDE
 - <http://selenium-ide.openqa.org>
- Selenium RC
 - <http://selenium-rc.openqa.org>
- Selenium Grid
 - <http://selenium-grid.openqa.org>
- Selenium XML
 - <http://sourceforge.net/projects/seleniumxml>



Q & A

ApacheCon



Leading the Wave
of Open Source

Contact Information

Brett G. Palmer

palmerb@automationgroups.com

