

Contents

Introduction	17
Welcome	17
Getting the Source-code	18
Windows, Linux or Mac	18
Typographical Conventions	19
Keeping Up To Date	19
Acknowledgements	20
1 Getting Started With The Tools	21
1.1 Selenium IDE	21
1.2 Java	22
1.3 Eclipse	22
1.4 Selenium RC	23
1.5 Summary	23
2 Selenium IDE Basics	25
2.1 How To Download and Install Firefox	25
2.2 How To Install The Selenium IDE	26
2.3 Capture Playback - Recording a Script With Selenium IDE	28
2.3.1 Command, Target and Value	29
2.3.2 Save The Test	29
2.3.3 Further reading	30
3 How To Install Java	33
3.1 Is Java already installed?	33
3.2 Install Java	33
4 Install and run Selenium RC	35
4.1 How To Download and Install Selenium RC	35
4.1.1 What to Download?	35
4.1.2 Download from Selenium HQ	36
4.1.3 Downloading Selenium 2	36

4.1.4	Downloading Selenium 1.0.3	37
4.1.5	Downloading Selenium 0.9.2	38
4.1.6	Downloading Troubleshooting	39
4.2	Overview of the Contents of the Selenium Archive	39
4.2.1	Archive Contents for 2.0	39
4.2.2	Archive Contents for 1.0.3 and 0.9.2	40
4.3	How to Work With Windows Command Line	41
4.3.1	Give Your Command Window a Title	41
4.3.2	An Alternative Command Line Tool	42
4.4	How to Open Command Windows At A Specific Folder	42
4.4.1	On Windows 7 and Vista	42
4.4.2	On Windows XP	42
4.5	How to start Selenium Server From Command Line	43
4.5.1	Generic Instructions	43
4.5.2	Selenium 2.0	44
4.5.3	Selenium 1.0.3	45
4.6	How to Stop the Selenium Server	45
4.6.1	From the command line	45
4.6.2	From a browser URL	45
4.7	How to Run IDE Tests In Different Browsers	46
4.7.1	End Notes	50
5	The Eclipse IDE	51
5.1	How to Install and Run The Eclipse IDE	51
5.1.1	Install Eclipse	51
5.1.2	Run Eclipse	52
5.2	How to Create a New Java project	52
6	Create a JUnit Test Using Selenium IDE	55
6.1	Overview of Exporting Section	55
6.2	Export an IDE script as a JUnit test	55
6.2.1	Load an existing script into the IDE	55
6.2.2	Formatting and Exporting using the IDE	56

6.2.3	Create a New Class in Eclipse	57
6.2.4	Copy and Paste the code generated by Selenium IDE into Eclipse	59
6.2.5	Resolve Import Errors	62
6.2.6	Add JUnit to the build path if necessary	64
6.2.7	Fix the Warning Messages	65
6.3	Run the JUnit test	67
6.3.1	Allow it through the Firewall	67
6.3.2	Seeing the test running	68
6.3.3	Check the results	69
6.4	It went too quickly!	70
6.4.1	Run the test in debug mode	71
6.5	Create and Import some more IDE Converted tests	74
6.6	Exercises	76
7	The Annotated Generated Test	77
7.1	Use Attach Source to see the Selenium Driver Source Code	77
7.2	Why should we do this?	79
7.3	How to find line numbers in Eclipse?	79
7.4	MyFirstSeleniumTests.java Annotated	80
7.5	A little about JUnit	86
8	Let's get coding	87
8.1	A brief pause, because you have already learned some Java	87
8.2	What we will now learn	87
8.2.1	Create a New Test From Scratch	88
8.2.2	@Test Annotation Turns Methods Into Tests	88
8.2.3	Back to Our Selenium Test	90
8.2.4	Making Something Happen	92
8.2.5	Your Final Test Code	93
8.2.6	The DefaultSelenium constructor explained	94
8.3	Retrospectives	95
8.3.1	Well Done	95
8.4	Why did you do that?	96

8.4.1	Run Selenium RC in a DOS window?	96
8.4.2	Stop and Restart Selenium Easily	97
8.5	Final Notes	97
8.6	Back to Eclipse	98
8.6.1	Removing the Source Code Association	98
9	What if it all goes wrong?	99
9.1	Check that Selenium Server is running	99
9.2	Close any Selenium RC windows	100
9.3	Stop and restart the Selenium RC server	100
9.4	Remove any Java threads from the Debug perspective	101
9.5	Close any blocked browser windows	102
9.6	Firefox Update Dialog	102
9.6.1	Summary	102
10	Essential Firefox Add-Ons	105
10.1	Firebug	105
10.1.1	What is Firebug?	105
10.1.2	Install Firebug	106
10.1.3	Basic Firebug Usage	106
10.2	FirePath For Firebug	108
10.2.1	What is FirePath?	108
10.2.2	Install FirePath	108
10.2.3	Basic Usage of FirePath	109
10.2.4	Optimise XPath with FirePath	110
11	Test Automation Thinking 101	113
11.1	Let's Automate Search	113
11.2	Start writing some tests	114
11.2.1	Make Selenium Type In Some Text	114
11.2.2	Fix the duplicate method name	115
11.2.3	Add a Selenium Command	115
11.2.4	The "type" method	116

11.2.5	Find the locator with Firebug	117
11.2.6	A brief introduction to XPath	117
11.2.7	Add the locator into the code	119
11.2.8	Check that it works	119
11.2.9	Now make it Click on the Search Button	120
11.2.10	Selenium “Click” Method	121
11.2.11	Locate the button with Firebug	121
11.2.12	Create an XPath so that selenium can locate it	121
11.2.13	Add the XPath details to our source code	121
11.2.14	One last thing to do	122
11.2.15	Now check that it works	122
11.2.16	Quick Summary	122
11.3	But is it a test if we don’t check any results?	123
11.3.1	But we can’t control the data	124
11.4	Acceptance Criteria: Selenium Remote Control homepage listed in re- turned results	125
11.4.1	Look for text in the page	125
11.4.2	Scan the page source code	129
11.4.3	Look for a specific URL in the page	131
11.4.4	Most Robust Method	135
11.5	Acceptance Criteria: Page title has “Selenium-RC” in it	135
11.6	Acceptance Criteria: Search box on the page has Selenium-RC in it	136
11.7	One test to rule them all	137
11.8	Summary	138
12	First Steps with Automation Refactored	139
12.1	Introducing Refactoring	139
12.2	Automatically refactoring the start up code into a new method	140
12.3	Automatically refactor the tear down code into a new method	142
12.4	JUnit @Before and @After	143
12.4.1	Refactoring Plan and Analysis	144
12.4.2	Remove parameter from stopSeleniumAndCloseBrowser	145
12.4.3	Move setup code into a new method	145

12.4.4	Remove the setup code from each @Test method	146
12.4.5	Annotate stopSeleniumAndCloseBrowser with @After	147
12.4.6	Remove the tear down code from each @Test method	147
12.4.7	The Refactored Code	148
12.4.8	@BeforeClass & @AfterClass	149
12.5	Automatically create JUnit Test Structure	152
12.6	Summary	154
13	How to upgrade Selenium	155
13.1	First Download the New Version	155
13.2	Configure Eclipse Project	155
13.3	Remember to re-attach the source	156
14	Basic HTML Theory	157
14.1	Page Structure	157
14.2	Elements & Tags	158
14.3	Attributes	158
14.4	Expanded	158
15	Basic XPath Theory	159
15.1	XPath Expressions	159
15.2	Node Types	160
15.3	Selections	160
15.3.1	Start from root	160
15.3.2	Start from Anywhere	160
15.3.3	By Element Attributes	161
15.4	Predicates	161
15.4.1	Predicates can be indexes	161
15.4.2	Predicates can be attribute selections	161
15.4.3	Predicates can be XPath functions	162
15.4.4	Predicates can be comparative statements	162
15.5	Combining Match Queries	162
15.6	Wild Card Matches	162

15.6.1	node()	162
15.6.2	*	163
15.7	XPath Functions	163
15.7.1	contains()	164
15.7.2	starts-with()	164
15.7.3	Many More	165
15.8	Boolean Operators	165
15.8.1	and	165
15.8.2	or	165
15.9	XPath optimisation	166
15.9.1	Use the ID	166
15.9.2	Use the attributes	166
15.9.3	Start at the first unique element	166
15.10	Selenium XPath Usage	167
16	Basic CSS Selector Theory	169
16.1	FirePath for Firebug	169
16.2	CSS Selector Expressions	169
16.3	Selenium and CSS	169
16.4	Selections	171
16.4.1	Direct Descendents and Absolute Paths	171
16.4.2	Sub Elements and Relative Paths	171
16.5	Attribute and Index matching	171
16.5.1	Attribute Matching	171
16.5.2	Special attribute selectors	172
16.5.3	Indexed Matching	172
16.6	Advanced	174
16.6.1	Combining Matches	174
16.6.2	Wild Card Matches	174
16.6.3	Attribute Substring matching	175
16.6.4	Boolean Operators	175
16.6.5	Sibling Combinators	176
16.7	Useful Links	176

17 Learning The Selenium API	179
17.1 Tips for Learning	179
18 Testing HTML Forms	181
18.1 An HTML Form	182
18.2 Submit a Form with Default Values	183
18.2.1 Commands Used	185
18.2.2 Submit Form without clicking button	186
18.3 Amend a Text, Password, TextArea or File Field	186
18.3.1 Text Fields	186
18.3.2 type vs typekeys	187
18.3.3 Password Fields	187
18.3.4 TextArea Fields	187
18.3.5 File Field	189
18.4 Amending Checkboxes and Radio Items	190
18.4.1 Click	191
18.4.2 Check	191
18.4.3 Uncheck	192
18.5 Amending Multiple Select Values	193
18.6 Amending Dropdown Elements	195
18.7 Amending Hidden Field Values	196
18.8 Checking the Values of the Input Fields	197
18.8.1 Text, Password, TextArea	197
18.8.2 CheckBox & Radio Items	198
18.8.3 Multiple Select & DropDown	199
19 Testing Static HTML	205
19.1 A Basic Static HTML Test	206
19.1.1 getTitle - Check the page title	207
19.1.2 getText - Check the username	209
19.1.3 isTextPresent - check the heading	210
19.1.4 isElementPresent - check comments and filename	211
19.1.5 getXPathCount - count the checkboxes entered	212

19.1.6	click - navigate to the form	213
19.1.7	The rest of the test	213
19.1.8	The full test and summary	214
19.2	getAttribute	215
19.3	assignId	216
19.4	getBodyText and getHtmlSource	217
19.5	goBack	218
19.6	highlight	219
20	Using JavaScript with Selenium	221
20.1	A Basic JavaScript Enabled Page	221
20.2	The Class	222
20.3	FireEvent, isAlertPresent, and getAlert	223
20.4	getEval	224
20.5	runScript	225
20.6	addScript and removeScript	226
21	Starting Selenium From Code	227
21.1	Add Selenium Server to the project	227
21.1.1	For Selenium 2.0.0 and above	227
21.1.2	For Selenium 1.0.3 and below	227
21.2	Start it in the code	228
21.3	The Start Up routine explored	229
21.3.1	BindException Handling	229
21.3.2	Stop Existing Server	229
21.3.3	Custom Remote Control Configurations	231
22	Running Tests Outside Eclipse	233
22.1	What is Ant?	233
22.2	What is Hudson?	233
22.3	What is Subversion?	234

23 Using Ant to Run Selenium Tests	235
23.1 Prerequisite: Install Java JDK	235
23.1.1 Install Java JDK	235
23.1.2 Update the Environment Variables and Path For Java	236
23.1.3 Check that Java works	237
23.2 Install Ant	237
23.2.1 Install Ant Automatically	238
23.2.2 Install Ant Manually	238
23.3 Check that Ant works	239
23.4 Create a Lib Folder	240
23.5 Amend Eclipse Project Properties	241
23.6 Create an initial build.xml file	241
23.7 Running the Ant file	249
23.8 Add the Tests which require no server	250
23.9 Refactor The Build File	251
23.10 Summary	255
23.11 Additional Reading	256
24 Using Hudson to Run Ant	259
24.1 Introducing Hudson	259
24.2 Install Hudson	259
24.3 Using Hudson without Version Control	261
24.4 Configure the Hudson Project	264
24.5 Scheduling the build automatically	267
25 Adding Subversion to the mix	269
26 Take Stock of Where we are	271
26.1 Plan Of Action	272
27 Evolving A Selenium Manager Class	275

28 Create Our First Page Objects	283
28.1 Create a SearchPageTests class	283
28.2 Add the SeleniumManager	284
28.3 Convert startSeleniumAndSearchForSeleniumRC	285
28.4 typeInASearchTermAndAssertThatHomePageTextIsPresent	290
28.5 typeInASearchTermAndAssertThatHomePageURLExists	291
28.6 typeInASearchTermAndCheckPageTitleHasSearchTermInIt	293
28.7 typeInASearchTermAndCheckSearchInputHasSearchTermInIt	293
28.8 After all that	294
28.8.1 SearchPageTests.java	294
28.8.2 SearchPage.java	295
28.9 Final Notes	296
29 More Page Object Creation	297
29.1 The Tests	297
29.1.1 HTML_form_tests.java	297
29.1.2 Static_HTML_tests.java	306
29.1.3 JavaScript_With_Selenium_Tests.java	312
29.2 The Page Objects	315
29.2.1 BasicHTMLForm.java	315
29.2.2 HTMLFormResultsPage.java	319
29.2.3 BasicAjaxPage.java	321
30 Page Object Models Summary	323
30.1 What is a Page Object Model?	323
30.2 Grow an abstraction layer	323
30.3 Refactor the Page Objects internally	324
30.4 Create Domain Level Abstractions	324
30.5 Lessons learned	325
30.6 Related Study	326

31 Data Driven Tests in JUnit	327
31.1 Basic Data Driven Testing	327
31.2 JUnit parameterized Class Runner And Constructor	328
31.3 Data supplied by an @parameters annotated method	329
31.4 A Data Driven Test	329
31.5 Running a Data Driven Test	330
31.6 The full code for the test	330
31.7 Multiple Tests	331
31.8 Reading Data From Tab Delimited Files	332
31.9 General Hints	334
32 Screen Capture on Failure	335
33 Run the tests on multiple browsers	341
33.1 Browser Codes	341
33.2 Property Files	343
33.3 System Properties	345
34 JavaScript And Dynamic HTML Testing Redux	349
34.1 Example Dynamic HTML Page	349
34.2 First Model the behaviour in the test	350
34.3 Code the check in the page object model	351
34.3.1 First understand the page	351
34.4 Summary	352
35 Cookie Handling	353
35.1 Cookies in the example application	353
35.2 How to see cookies on a web page?	353
35.2.1 Firefox by Default	354
35.2.2 Firecookie - Firebug Plugin	354
35.3 Selenium Cookie Commands	356
35.4 Example Code For Cookie Testing	357
35.4.1 Basic Class	357

35.4.2	Check Cookie Creation using deleteAllVisibleCookies and is-CookiePresent	358
35.4.3	Get Cookie values using getCookie and getCookieByName	359
36	The Future of Selenium	365
36.1	If you want to experiment with WebDriver	366
37	Structuring the tests and code	369
37.1	Java project folder and package structures	370
37.2	Changing what goes into Version Control	372
37.2.1	Configuring Source-code Folders in Eclipse	372
37.2.2	Add libraries	373
37.3	Re-organise the Classes and Packages	373
37.4	Share the Server among tests and create a base test class	375
37.4.1	Share the Server using a Singleton	375
37.4.2	Create a Base Class	376
37.5	Get Ant Working	379
37.6	Make sure all tests are valid tests	380
37.6.1	Create end to end tests where we assert after executing functionality	381
37.6.2	Have all tests pass on multiple browsers	381
37.6.3	Locators as constants or methods	382
37.7	Get Hudson Working	384
37.8	Speed up the test execution with a suite	387
37.9	Running Different Suites for Different Browsers	392
37.9.1	Nested Suites	392
37.9.2	Browser Specific Groups	393
37.9.3	Create a conditional Ant File	394
37.10	Get WebDriver working	396
37.11	Ways of using what we have built	396
37.12	Exercises For the reader:	397
38	In Closing	399
38.1	Additional Reading	399

A	Playing along at home	403
A.1	Prerequisites	403
A.2	Get The Code	403
A.2.1	From A Zip Archive	403
A.2.2	From the Subversion Repositories	404
A.3	Create a Project	404
B	The SeleneseTestCase Class	405
B.1	What is SeleneseTestCase	405
B.2	Details	405
B.3	A look at SeleneseTestCase.java	405
B.4	Use the SeleneseTestCase Functionality	407
B.4.1	Using the IDE to add asserts and verifies	407
B.4.2	What is the difference between an Assert and a Verify?	409
	Index	414
	About the Author	417