

## ABSTRACT

This paper introduces Web Application Testing System “Selenium, its core and other components as well as its features. Also, covers the answers to some very important questions like; is automation always advantageous? When should one decide to automate test cases?

The paper also compares Selenium tool with other available testing tool and its pros and cons over them with keeping in mind the needs of modern day websites.

## INTRODUCTION

Many, perhaps most, software applications today are written as web-based applications to be run in an Internet browser. The effectiveness of testing these applications varies widely among companies and organizations. In an era of highly interactive and responsive software processes where many organizations are using some form of Agile methodology, test automation is frequently becoming a requirement for software projects. Test automation is often the answer. Test automation means using a software tool to run repeatable tests against the application to be tested. For regression testing this provides that responsiveness.

Considering the needs of modern day websites, different testing tools are no more an option but they are often supposed to be a regular part of testing cycle. Technological advancement and competition is so fierce, that adding more and more human resources in testing team, to keep up with the pace of development activities is not always an optimized solution to the problem.

Sometimes its integrity of the application due to code changes, sometimes application verification across multiple OS or web browsers, and sometimes simulation of certain business logic scenarios that make the “efficient-test-automation” a basic need of the day, rather than just some usual option in the testing life cycle.

Test automation has specific advantages for improving the long-term efficiency of a software team’s testing processes. Test automation supports:

- Frequent regression testing
- Rapid feedback to developers
- Virtually unlimited iterations of test case execution
- Support for Agile and extreme development methodologies
- Disciplined documentation of test cases
- Customized defect reporting
- Finding defects missed by manual testing

There are many advantages to test automation. Most are related to the repeatability of the tests and the speed at which the tests can be executed. But it is **not** always advantageous to automate test cases. There are times when manual testing may be more appropriate. For instance, if the application’s user interface will change considerably in the near future, then any automation might need to be rewritten anyway. Also, sometimes there simply is not enough time to build test automation. For the short term, manual testing may be more effective. If an application has a very tight deadline, there is currently no test automation available, and it’s imperative that the testing get done within that time frame, then manual testing is the best solution.

There are a number of commercial and open source tools available for assisting with the development of test automation. Selenium is possibly the most widely-used open source solution. Selenium comes as one of the popular answers to the HOT question of Test Automation these days. For many obvious reasons, Selenium is enjoying spotlight in its domain.

Selenium is set of different software components each with a different approach to supporting test automation. Most Selenium QA Engineers focus on the one or two components/ tools that most meet the needs of their project, however learning all the tools will give you many different options for approaching different test automation problems. The entire suite of tools results in a rich set of testing functions specifically geared to the needs of testing of web applications of all types. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behavior. One of Selenium’s key features is the support for executing one’s tests on multiple browser platforms.

## Selenium Components

Selenium has 3 major components. These components are also sequential usage pattern of Selenium in general. Let's take a look at logical definition of these components.

### 1. Selenium IDE (Integrated Development Environment)

#### Start-up: write and run tests in Firefox

Selenium IDE is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension, and allows you to record, edit, and debug test scenarios. Selenium IDE includes the entire Selenium Core, allowing you to easily and quickly record and play back tests in the actual environment that they will run by generating the recording code in Selenium supported languages, and provides an interface to debug code elements.

Selenium IDE gives an edge to other automation tool with its recording capability. It is not only recording tool: it is a complete IDE so, you can choose to use its recording facility, or you may edit your scripts by hand.

With auto complete support and the ability to move commands around quickly, Selenium IDE is the ideal environment for creating Selenium tests no matter what style of tests you prefer.

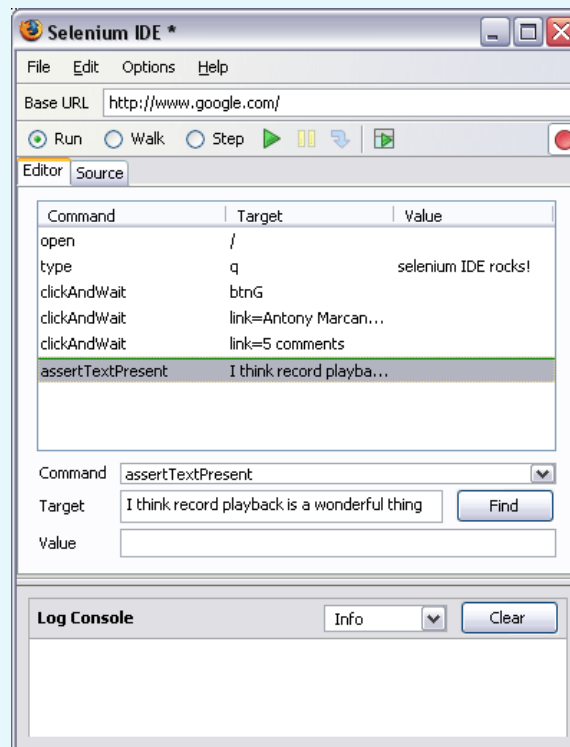


Figure 1

There are number of features associated with Selenium IDE, some of them are as under:

- Easy record and playback
- Intelligent field selection will use IDs, names, or XPath as needed
- Auto complete for all common Selenium commands
- Walk through tests
- Debug and set breakpoints
- Save tests as HTML, Ruby scripts, or any other format
- Support for Selenium user-extensions.js file
- Option to automatically assert the title of every page
- Easy customization through plug-ins

As of 1.0.4, Selenium IDE has had a plug-in system to allow for easy extension and customization including:

- Adding new functionality to the API
- Changing existing functionality
- Custom formats and export capabilities
- Hosting of plugin update.rdf files
- Adding new locator strategies

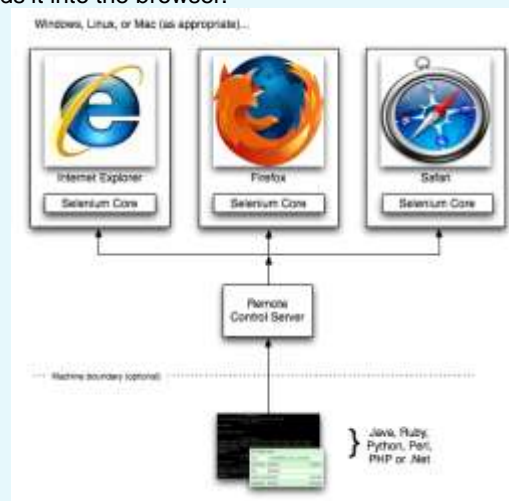
## 2. Selenium RC (Remote Control) Customization: language/ browser

Selenium RC comes in two parts:

- A server which automatically launches and kills browsers, and acts as a HTTP proxy for web requests from them.
- Client libraries for your favorite computer language.

**Selenium Remote Control (RC)** allows running automation test in multiple browsers and platforms as well as in any specific code environment e.g. Eclipse IDE for JAVA, or Visual Studio for C#. Tweak user tests in their preferred language by adding programming support to enhance the functionality of an automation script.

Selenium Remote Control (RC) is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser. The RC server also bundles Selenium Core, and automatically loads it into the browser.



Selenium Remote Control is great for testing complex AJAX-based web user interfaces under a Continuous Integration system. It is also an ideal solution for users of Selenium Core or Selenium IDE who want to write tests in a more expressive programming language than the Selenese HTML table format customarily used with Selenium Core.

### 3. Selenium Grid Deploying: scale out and speed up

Selenium Grid is an extension to Selenium RC that dramatically speeds up functional testing of web-apps by leveraging the existing computing infrastructure. It allows to easily run multiple instances of one automation script or complete suite of automation tests in parallel, on multiple machines, in an heterogeneous environment, cutting down the time required for running in-browser test suites. Selenium grid is helpful in time saving by distributing the tests across multiple servers and running tests in parallel.

There are several tremendous advantages of Selenium Grid. Some of them are:

- Tests can be ran in parallel for reduced execution time
- Test can be ran on various browser and operating system combinations
- Generates a collated report from all test machines
- The tests are distributed by the Hub server to multiple Selenium RC machines
- The tests can be ran by various Cloud computing services such as Amazon EC2, Sauce Labs, BrowserMob, and PushToTest.

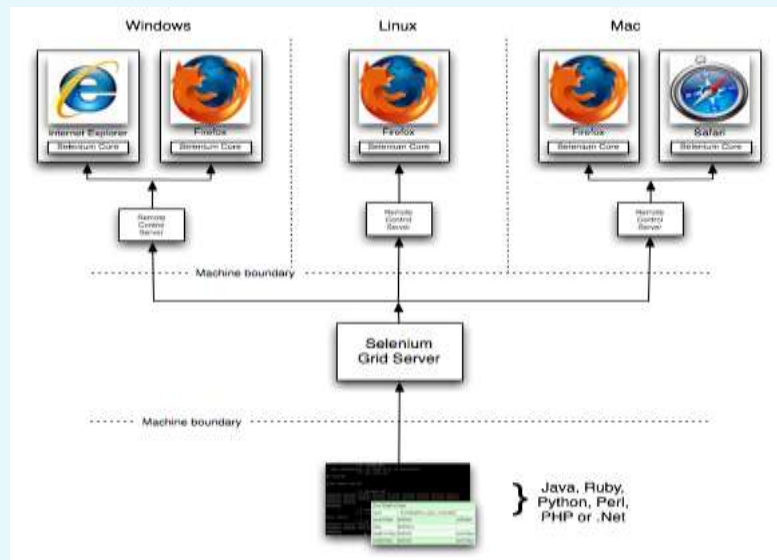


Figure 2

### Other Components

#### 4. Selenium Core

Selenium Core is the DHTML test execution framework and an original JavaScript-based testing system. It's now used primarily as a component of Selenium Remote Control, but it can also be used as a pure JavaScript/HTML testing system. It is the engine of both, Selenium IDE and Selenium RC (driven mode), but it also can be deployed on the desired application server.

## 5. Selenium on Rails

Selenium on Rails provides an easy way to test Rails application with Selenium Core. Suite files are created on the fly for each directory and the Selenium Core files don't have to pollute /public. It can support document updating but editing the test cases is currently considered out of scope for this plug-in.

This plug-in does four things:

- i. The Selenium Core files don't have to pollute /public.
- ii. No need to create suite files, they are generated on the fly e.g. one suite per directory in /test/selenium (suites can be nested).
- iii. Instead of writing the test cases in HTML you can use a number of better formats (see Formats).
- iv. Loading of fixtures and wiping of session (/selenium/setup).

## 6. Selenium on Ruby

Selenium on Ruby is the hub for newer Ruby related Selenium projects. The current picture of Selenium on Ruby is composed of several selenium related ruby gems that work together.

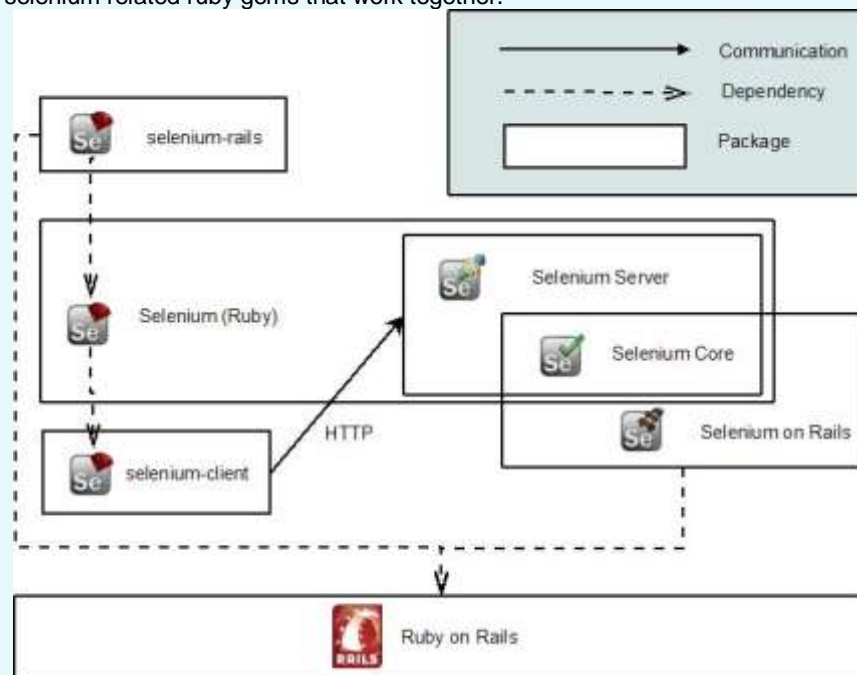


Figure 3

**Selenium Client** is the second and better release of Ruby driver for Selenium RC. It provides a simple yet idiomatic Ruby API to write Selenium tests in Ruby. It allows a straightforward way to write Selenium tests in Ruby as well as additional support for rSpec and rake.

**Selenium Ruby** bundles Selenium RC Server to provide additional server management.

**Selenium Rails** supports the use of Selenium RC with Ruby on Rails projects.

## 7. CubicTest for Eclipse

It's a graphical Eclipse plug-in for writing Selenium and Watir tests. CubiTest makes web tests faster and easier to write, and provides abstractions to make tests more robust and reusable. It is an editor which is centered around pages/states and transitions between these pages/states.

The model is intuitive for both Ajax and traditional web applications and supports most user interaction types. It features an innovative test recorder and test runner based on Selenium RC which are fully integrated with the graphical test editor.

## 8. CubicTest for Eclipse

Bromine is a web-based QA tool for selenium, which enables you to easily run selenium RC tests and view the results. It scales beautifully from the single tester that just wants to run some tests without all the hassle to the corporate solution with multiple user-groups and hundreds of test cases.

Bromine is a web-based QA tool for Selenium that enables you to easily run Selenium-RC tests and view the results. It scales beautifully from the single tester that just wants to run some tests without all the hassle to the corporate solution with multiple user-groups and hundreds of test cases.

Bromine can be used for two main reasons:

- i. To provide an easy way to run Selenium tests and store the results for analysis and backlog
- ii. To present the results in a way that less technical minded management would understand

Features of Bromine are:

- Supports tests in Java and PHP
- Record tests with provided IDE formats
- Upload tests
- Run multiple tests against multiple OS/browser combinations with a single click
- Setup OS/browser testing needs with checkboxes
- Uses completely unmodified RC servers
- No RC server hassle, just plug in the IP and you're set
- Seamless load balancing. If you have multiple RC servers defined, they will be used (not GRID, our own solution)
- Full log of results. All commands executed and their status are stored
- Define your own user groups and their rights
- Coded in CakePHP, MVC design, accessible API, easily extendable
- Add plug-ins easily

## Why choose Selenium?

Selenium is free having no licensing costs and no maintenance renewal costs, so it can be deployed to as many machines as needed in a lab or for local development.

The Selenium client libraries can be imported into most popular language IDE's i.e. Eclipse, NetBeans, Visual Studio .NET, etc. It reuses existing unit testing frameworks. The Selenium is best integrated into Fitness frameworks and rally round the integration into the build process by making it easy to deploy into Continuous Integration Environments. It has strong support for AJAX technologies and provide Cross platform browser and operating system comparison testing.

The QA test teams can use it for functional, regression, and user acceptance testing (UAT). The developers can also use selenium for test driven development (TDD) in the Agile and extreme programming (XP) community. Moreover, the developers can also use Selenium into other areas like;

- Conversion into production monitoring and load testing
- Extended hooks into Flash, Flex, and other popular technologies

It also supports run tests in parallel with the Grid tests across multiple RC (agent) machines from a single Hub machine which will save time, and runs faster.

Selenium uses cloud computing services for; Amazon EC2 (functional, regression testing), Sauce Labs (functional, regression testing), BrowserMob (production monitoring, load testing) and PushToTest (production monitoring, functional testing, load testing). It opts to have a remote test lab if hardware is limited. The framework is simple and the code is neat and maintainable.

## Weaknesses of Selenium

The major drawback of selenium is that it's a testing tool for web based applications only. It has no result reporting engine of its own and user has to use integrating tools for that. Do not support database checks and test/error recovery. Selenium has lack of support documentation at certain topics. And do not allow keyword driven testing.

### Comparison with Automated Testing Tools

	Silk Test	Watir	Selenium	Pure Test	MaxQ	WET
<b>Record &amp; Playback</b>	Yes	No	Yes	Yes	Yes	Yes
<b>Web Testing</b>	Yes	Yes	Yes	Yes	Yes	Yes
<b>Browser Support</b>	IE 6.0 Fire fox 1.5.0.1+ Netscape Navigator AOL 9 and 9	Only IE	Internet Explorer 6.0 Firefox 0.8 to 1.5 Mozilla Suite1.6 Opera 8	Only IE	IE, Fire fox	Only IE
<b>Ease of Use</b>	Good	Easy	Easy	Good	Good	Good
<b>Database Tests</b>	Yes	Yes	No	No	No	Yes
<b>Object Mapping</b>	Yes	No	No	No	No	Yes
<b>Object Identity Tool</b>	No	No	No	No	No	Yes
<b>Extensible Language</b>	Yes	Yes	Yes	No	No	Yes
<b>Integration</b>	Yes	Yes	Yes	No	No	Yes
<b>Image Testing</b>	Yes	No	Yes	No	No	Yes
<b>Test/Error Recovery</b>	Yes	Yes	No	No	No	Yes
<b>Scripting Language</b>	4Test	Ruby	Supports writing tests in Java, Perl, Python, Ruby	Custom	Custom	Ruby
<b>Installation/ Setup</b>	Easy	V Good One click installers	Ok but requires setup of proxy server	Easy but requires a proxy server	Easy	Easy
<b>Documentation</b>	Good	V Good	Good	V Good	Good	Ok

## ENDING NOTES

Selenium is a test tool for web applications. Selenium tests run directly in a browser, just as real users do. And they run in Internet Explorer, Mozilla and Firefox on Windows, Linux, and Macintosh. No other test tool covers such a wide array of platforms.

We can refer to Selenium as an effective web automation tool with several options to create flexibility of use and application. Limitations and issues apart, Selenium does many of required activities and fulfils the needs of most of the software developers and testing teams. If Selenium team focuses on current technical problems in Selenium framework and documentation support, then it can dominate the automation testing market with more strength and grip. Till then, Selenium exists with a good competition of similar domain tools and makes a name for it.

## REFERENCES

1. <http://www.openqa.org/selenium/>
2. <http://seleniumhq.org/about/platforms.html>
3. <http://www.slideworld.com/slideshow.aspx/Selenium-Testing-ppt-2767913>
4. <http://www.qualitytesting.info/forum/topics/how-selenium-remote-control?page=1&commentId=2064344%3AComment%3A103902&x=1#2064344Comment103902>