**What is selenium?** 

Selenium is a suite of tools to automate web browsers across many platforms.

Suite of Tools:

Selenium IDE

Selenium RC

Selenium WebDriver

Selenium Grid

Selenium 1 = IDE + RC + Grid

Selenium 2 = IDE + RC + Web Driver + Grid

Selenium...

Runs in many browsers and operating systems.

Controlled by many programming languages and testing frameworks.

**History:**

* Selenium first introduced in 2004 by Jason Huggins at Thought Works.
* WebDriver introduced in 2006 by Simon Stewart at Google.
* The both Selenium RC and WebDriver were merged in 2008.

JavaScript library that could drive interactions with the page, allowing to automatically rerun tests against multiple browsers and the library is named as **Selenium Core**. **Selenium Core**, which underlies all the functionality of Selenium Remote Control (RC) and Selenium IDE. **Selenium RC** is the first tool to control a browser from a language of your choice.

**Merge of Selenium and WebDriver:**

“Why are the projects merging? Partly because WebDriver addresses some shortcomings in selenium (by being able to bypass the JS sandbox, for example. And we’ve got a gorgeous API), partly because selenium addresses some shortcomings in WebDriver (such as supporting a broader range of browsers) and partly because the main selenium contributors and I felt that it was the best way to offer users the best possible framework.”

**Selenium IDE**

It and allows for recording, editing, and debugging of functional tests. It was previously known as Selenium Recorder. Selenium-IDE was originally created by Shinya Kasatani and donated to the Selenium project in 2006. Selenium IDE was previously little-maintained. Selenium IDE began being actively maintained in 2018.

Scripts are recorded in Selenese, a special test scripting language for Selenium. Selenese provides commands for performing actions in a browser (click a link, select an option), and for retrieving data from the resulting pages.

**Selenium IDE features:**

• Create Test Cases, Test suites (We can Record test cases or type Test steps using element locators and Selenese commands)

• Edit Test Cases

• Execute Test cases, Test suites

• Debug Test Cases.

• Enhance Test Cases

• Export Test cases to other formats (java, ruby etc...) **[**Older version of IDE**]**

**Drawbacks of Selenium IDE**

• It supports Mozilla Firefox and Chrome (with latest IDE) browser only.

• It doesn't support Programming logic/features to enhance Test cases.

• It doesn't support Data Driven Testing.

• It is not suitable for complex test case design.

• No centralized maintenance of Objects/Elements

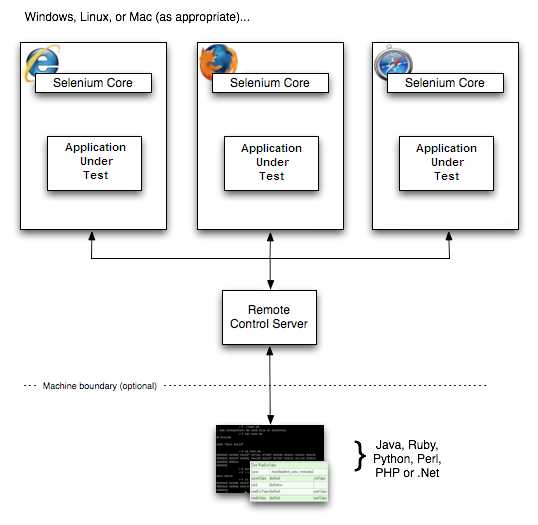
**Selenium Remote Control:**

Selenium RC components are:

The Selenium Server which launches and kills browsers, interprets and runs the Selenese commands passed from the test program, and acts as an HTTP proxy, intercepting and verifying HTTP messages passed between the browser and the AUT.

Client libraries which provide the interface between each programming language and the Selenium RC Server.

**Architecture diagram….**



The diagram shows the client libraries communicate with the Server passing each Selenium command for execution. Then the server passes the Selenium command to the browser using Selenium-Core JavaScript commands. The browser, using its JavaScript interpreter, executes the Selenium command. This runs the Selenese action or verification you specified in your test script.

**Selenium Server**

Selenium Server receives Selenium commands from your test program, interprets them, and reports back to your program the results of running those tests.

**Client Libraries**

The client libraries provide the programming support that allows you to run Selenium commands from a program of your own design. There is a different client library for each supported language. A Selenium client library provides a programming interface (API), i.e., a set of functions, which run Selenium commands from your own program. Within each interface, there is a programming function that supports each Selenese command.

**The Same Origin Policy Issue**

**Same Origin policy prohibits accessing the JavaScript code from one domain to another domain**. Example, the HTML code in www.google.com uses a JavaScript program "randomCaptchaGeneration.js". The same origin policy will only allow **randomCaptchaGeneration.js** to access pages within google.com such as signup, login, logout, inbox, sent and others. But, it cannot access pages from different sites such as yahoo.com/search or facebook.com since they belong to different domains.

**Selenium WebDriver:**

Communicates directly with browser without any server [like in RC]

HTMLUnit driver without launching the browser.

Realistic and faster Execution, since they are directly interacting with Browser

Browser driver varies from one browser to another.

Selenium-WebDriver makes direct calls to the browser using each browser’s native support for automation. How these direct calls are made, and the features they support depends on the browser you are using.

The same RC languages are supported with WebDriver

Java

C#

PHP

Python

Perl

Ruby

Different Operating Systems:

Windows

MAC

UNIX

**Drawbacks:**

No built-in detailed report creation after execution.

Requires programming knowledge

No built-in mechanism for logging.

**Selenium Grid:**

Selenium Grid is used to execute tests across multiple browsers, operating environments and machines in parallel.

Why grid:

* To run your tests against multiple browsers, multiple versions of browser, and browsers running on different operating systems.
* To reduce the time it takes for the test suite to complete a test pass.

Selenium Grid 1 supports only Selenium RC Tests.

Selenium Grid 2 supports Selenium RC Tests as well as Selenium WebDriver Tests.

|  |  |
| --- | --- |
| **Selenium** | **QTP** |
| Open Source, Free to use | Commercial |
| Supports various operating systems | Only Windows |
| Executed without browser under visible | Applications to be visible while executing |
| Executes in parallel | Using Quality Center |
| Only web based applications | Client server, web applications and desktops |
| Support - User community | Support – Paid |
| Language - Java, JavaScript, Perl, Python and PHP | Language – VB script |

**References:**

<https://www.seleniumhq.org/docs/01_introducing_selenium.jsp#test-automation-for-web-applications>

<https://www.seleniumhq.org/about/history.jsp>

<https://www.seleniumhq.org/ecosystem/>

**Selenium Blog** <https://seleniumhq.wordpress.com/>

**Selenium** **IDE** <https://www.seleniumhq.org/docs/02_selenium_ide.jsp#chapter02-reference>

**Selenium** **RC** <https://www.seleniumhq.org/docs/05_selenium_rc.jsp#chapter05-reference>

**WebDriver** <https://www.seleniumhq.org/docs/03_webdriver.jsp#chapter03-reference>

**Selenium Grid** <https://www.seleniumhq.org/docs/07_selenium_grid.jsp#chapter07-reference>

**Google.com** ☺