

# Selenium GRID and Nunit

Learn Selenium Testing

Aditya

# Test Execution in VS Framework

.Visual Studio 2010.

.Go to 'Test View'-Right Click on test->select  
Execute/Debug Selection.

# Pros - Cons of IDE based execution

## .Pro:

- 1.Easy to find and select.
- 2.Easy to execute.
- 3.Very easy to debug and fix right away.
- 4.Easy tracking if tests is buggy.

## .Cons.

- 1.Slow test execution as only one test can execute at a time.
- 2.Less utilization of resources.
- 3.Difficult to handle visual studio for inexperienced user.

# Selenium Grid

- .Execute multiple tests in parallel.
- .Faster verification process.
- .The selenium-server-standalone package includes the Hub, WebDriver, and legacy RC needed to run the grid.
- .Ant is not required anymore.

# Setup

.Step 1: Start the hub.

.Step 2: Start the nodes.

.Step 3: Execute tests.

# Step 1: Start the hub

## HUB:

The Hub is the central point that will receive all the test request and distribute them the right nodes.

.Open a command prompt and navigate to the directory where you copied the selenium-server-standalone file. Type the following command:

```
java -jar selenium-server-standalone-2.14.0.jar -role hub
```

.The hub will automatically start-up using port 4444 by default. To change the default port, you can add the optional parameter `-port`.

```
-port 1234
```

. When you run the command. You can view the status of the hub by opening a browser window and navigating to: <http://localhost:4444/grid/console>

## Step 2: Start the nodes

.Regardless on whether you want to run a grid with new WebDriver functionality, or a grid with Selenium 1 RC functionality, or both at the same time, you use the same selenium-server-standalone jar file to start the nodes.

```
java -jar selenium-server-standalone-2.14.0.jar -role node -hub http://localhost:4444/grid/register
```

–Note: The port defaults to 5555 if not specified whenever the "-role" option is provided and is not hub.

```
java -jar selenium-server-standalone-2.14.0.jar -role node -hub http://localhost:4444/grid/register -port 5556
```

# Step3: Execute Test

.Now that the grid is in-place, we need to access the grid from our test cases. For the Selenium 1 RC nodes, you can continue to use the DefaultSelenium object and pass in the hub information.

```
Selenium selenium = new DefaultSelenium("localhost", 4444, "*firefox", "http://www.google.com");
```

.For WebDriver nodes, you will need to use the [RemoteWebDriver](#) and the [DesiredCapabilities](#) object to define which browser, version and platform you wish to use.

**We will use NUnit to execute test instead of Visual Studio.**

# NUnit.exe

nunit.exe, is a Windows Forms application that allows you to work selectively with your tests and provides graphical feedback.

<http://www.nunit.org/index.php?p=download>

# Demo

.Start Hub:

✓ Start hub as per step 1:Start the hub.

.Start number of nodes on different port.

✓ Start nodes as per step 2: start the node.

.Rebuild SeleniumProject start the test.

*C:*

*\selenium\SeleniumProject\SeleniumProject\bin\Debug\SeleniumProject.  
dll*

.Open that many instances of NUnit and load various project (\*.dll) build from pervious step.

.Select different test cases in different NUnit standalone and Run them.

Thank You!!

Questions?