

## Getting Started with ZBot and Selenium (JAVA) Integration

## Revision History

Version	Type	Description/Comment	Date	Author
1.0	Creation	Initial Release	6/24/11	Daniel Gannon

## Table of Contents

1 - Overview .....	4
2 - Required Software.....	5
2.1 Installing Java.....	5
2.2 Installing Mozilla Firefox.....	6
2.3 Installing ZBot.....	7
2.4 Installing Selenium IDE.....	8
2.5 Acquiring Selenium Server .....	9
2.6 Acquiring Selenium Client Driver .....	9
2.7 Acquiring Eclipse IDE .....	10
2.8 Installing JUnit Plug-in for Eclipse IDE (Optional) .....	10
3 - Using Selenium (JAVA) with ZBot.....	12
3.1 Selenium IDE.....	12
3.2 Eclipse IDE .....	14
3.3 Automation .....	17
3.3.1 Using the Command Line .....	17
3.3.2 Starting Selenium Server.....	18
3.3.3 Test Case Script.....	18
3.4 Running an Automated Test Case .....	19

## 1 - Overview

### How this guide will help

The “Getting Started with ZBot and Selenium (JAVA) Integration” guide will show you how to integrate Selenium automation test cases (Java based) with ZBots in order to return automation execution results back to Zephyr, typically through the following steps:

- Authoring test cases in Selenium and exporting them to Java (JUnit)
- Compiling these tests and generating batch scripts to call them through ZBot
- Executing these batch automated test case scripts from Zephyr

This guide assumes you already have a working knowledge of Selenium and Java. This guide will attempt to get you started by going over the required software, in “Required Software”, which are necessary to perform the tasks detailed in the second half of the guide, “Using Selenium (JAVA) with ZBot”. Versions of required software are subject to change, including Zephyr. It is recommended that you try to use the most up to date version of software listed, to ensure best compatibility with your system and security.

At the end of the document, sample code and output is provided to help get you started. After reading and understanding this guide, you will be able to perform the basic operations of integrating Selenium Java automation with Zephyr ZBots.

## 2 - Required Software



### 2.1 Installing Java

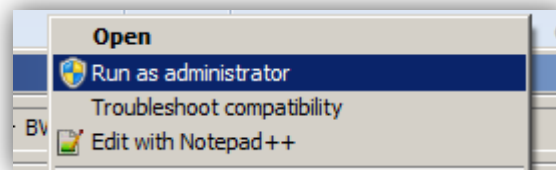
Before installing Java, make sure to uninstall all other unnecessary previous installations of Java before installing.

Java Download Site: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Installation Steps:

#### 1. Run the installer

Navigate to the directory where you downloaded the executable to. You must have administrative rights in order to install on Windows. This can be done by right-clicking the icon and then clicking "Run As Administrator" from the menu.



#### 2. Running with Java Plug-in

You may need to close browser applications and some various programs in order to complete the installation process. You will need to also restart browser applications in order to enable Java for them.

#### 3. Update the PATH and JAVA\_HOME variables (Optional)

You can run Java without setting the PATH and JAVA\_HOME variable or you can set it for convenience. Setting the PATH variable allows you to be able to conveniently run the executables from any directory without having to type the full command. Setting it this way will allow it to persist through reboot.

Setting up PATH in Windows (steps vary on Windows version):

1. Click **Start> Control Panel> System**
2. Click **Advanced> Environment Variables**
3. Scroll through the fields until you find PATH
4. Add the location of the bin folder of the install for PATH in System variables (also User variables if present).

An example of a typical path is: *C:\Program Files\Java\JDK\_1.6.0\_<version>\bin*

Setting up JAVA\_HOME in Windows (steps vary with Windows version):

1. Click **Start> Control Panel> System**
2. Click **Advanced> Environment Variables**
3. Scroll through the fields until you find JAVA\_HOME, if it's not present Add it to the System variables.
4. Add the location of the JDK root folder to the value of JAVA\_HOME in System variables (also User variables if present).

A typical path is: *C:\Program Files\Java\JDK\_1.6.0\_<version>*

#### Hints

- PATH variables are separated by a ";" are not case sensitive, and are read by Windows from left to right.
- Only one JDK variable is allowed
- Only applies to new command windows opened after committing

## 4. Installation Complete

You should now be ready to use Java on your machine.

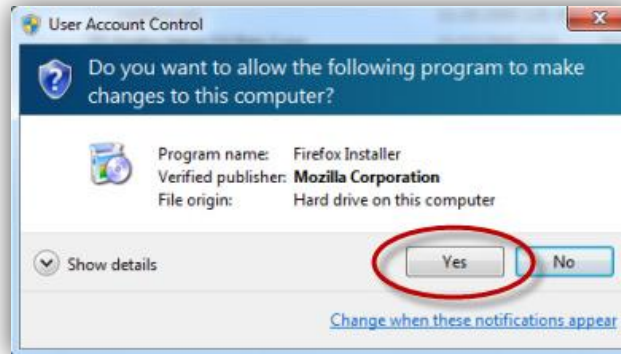
## 2.2 Installing Mozilla Firefox

Before installing a newer version of Mozilla Firefox, be sure to first uninstall any previous versions.

Firefox Website: <http://www.mozilla.com/en-US/firefox/fx/>

Upon visiting the Firefox website, you will be recommended the latest version compatible with your system. You may download the recommended version or another compatible version of your choosing.

**Windows Vista / 7** Users may see a User Account Control message. In this case, allow the setup to run by clicking Yes.



Upon successful completion of the installation, open Firefox and be sure everything works properly.

#### Hint

- Firefox Pop-up blocker may cause issues and is recommended that you disable it while testing Selenium functionality

## 2.3 Installing ZBot

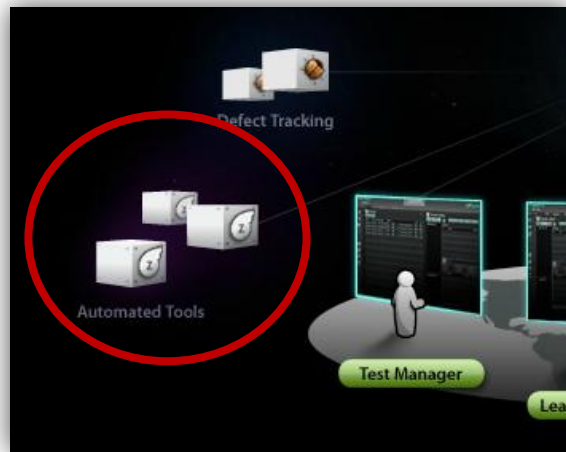
Zephyr Bots or ZBots are software agents that reside on target automation systems and form the link between the Zephyr Server and the automation tool. ZBot's default behavior is to accept commands from Zephyr and execute them on the target machine; the automation tools supported are ones that have support for command line kick off and parameter functionality.

ZBots default functionality can be extended upon by developing your own ZIP, a ZIP or a Zephyr Interceptor Program is an extension to the execution utility of a ZBot allowing for the default execution flow of automation scripts to be intercepted in order to perform customized actions. ZBots come packaged with a default ZIP which can be intercepted with a partial or full functionality customized. Currently, ZIPs written in JAVA are supported.

ZBot Download (Input your server name): <http://<Your Zephyr Server>/zephyr/zbot/>

Or

Select **Automated Tools** from Zephyr Desktop:



The installation setup file and detailed install instructions can be found from either location.

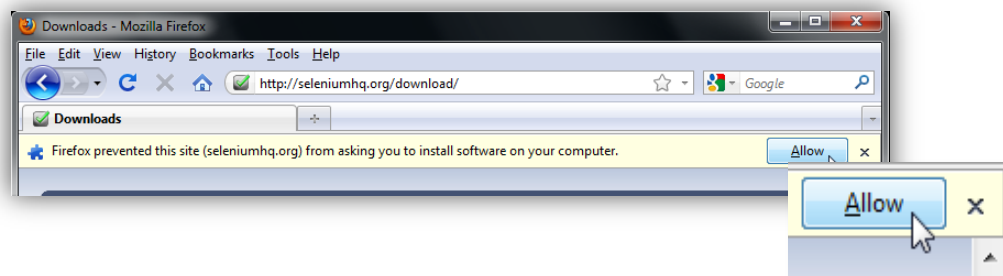
## 2.4 Installing Selenium IDE

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 tests. Selenium IDE includes the entire Selenium Core, allowing you to easily and quickly record, or edit scripts by hand, and play back tests in the actual environment that they will run.

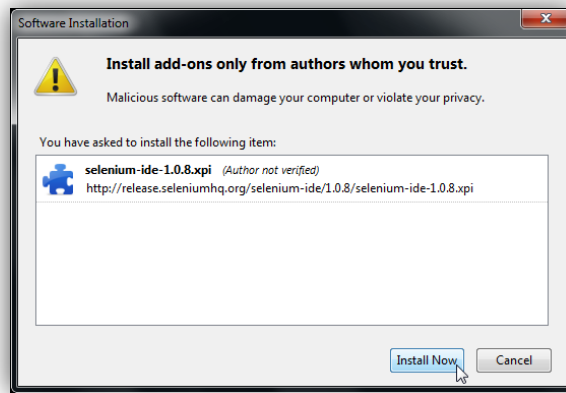
Selenium Download Site: <http://seleniumhq.org/download/>

Since Selenium IDE only runs on Firefox, you must browse to and install from the Selenium website within a compatible Firefox browser. It is recommended to install the latest version of Selenium IDE to avoid issues.

Using Firefox, first, download the IDE from the Selenium download site above. Firefox will protect you from installing add-ons from unfamiliar locations, so you will need to click Allow to proceed with the installation, as shown.



After allowing the add-on from Selenium, you will be prompt with another confirmation pop-up. This is the software installation window and details what you will be installing.



Click Install Now to begin the installation process and a progress bar will show you how the install is going. After the installation is finished, an Add-on pop-up window will appear telling you what add-on was installed and asking you to restart the browser. Upon restarting, your installation is complete. To confirm the existence of Selenium IDE on your machine, go to **Tools> Selenium IDE**.

For further instructions and feature explanation, visit:  
[http://seleniumhq.org/docs/02\\_selenium\\_ide.html#installing-the-ide](http://seleniumhq.org/docs/02_selenium_ide.html#installing-the-ide)

## 2.5 Acquiring Selenium Server

The Selenium Server is needed in order to run either Selenium RC style or Remote Selenium WebDriver scripts.

Selenium Download Site: <http://seleniumhq.org/download/>

Once you have the Selenium server JAR file downloaded, you may want to place it in a easy to access location for later use.

IE: `C:\selenium\selenium-remote-control\selenium-server-standalone-2.0rc2.jar`

## 2.6 Acquiring Selenium Client Driver

In order to create scripts that interact with the Selenium Server or create local Selenium WebDriver script you need to make use of language-specific client drivers.

Selenium Download Site: <http://seleniumhq.org/download/>

Once you have the Selenium client driver for JAVA downloaded, you may want to place it in an easy to access location for later use.

IE: `C:\selenium\client-driver\selenium-java-2.0rc2.jar`

## 2.7 Acquiring Eclipse IDE

Download from: <http://www.eclipse.org/>

Once downloaded, no installation is actually needed. Eclipse IDE can run immediately after you have finished downloading. It is recommended however, that you place the Eclipse folder in an easily accessible and secure place where it can't become compromised.

Upon first starting Eclipse IDE, you will be asked where you want to make your workspace. The workspace is another folder used by Eclipse to place all your projects, java files, and classes you will build. The workspace is not final. Once created, you can easily make a fresh new one (keeping the old one) or import a different one from another source.

## 2.8 Installing JUnit Plug-in for Eclipse IDE (Optional)

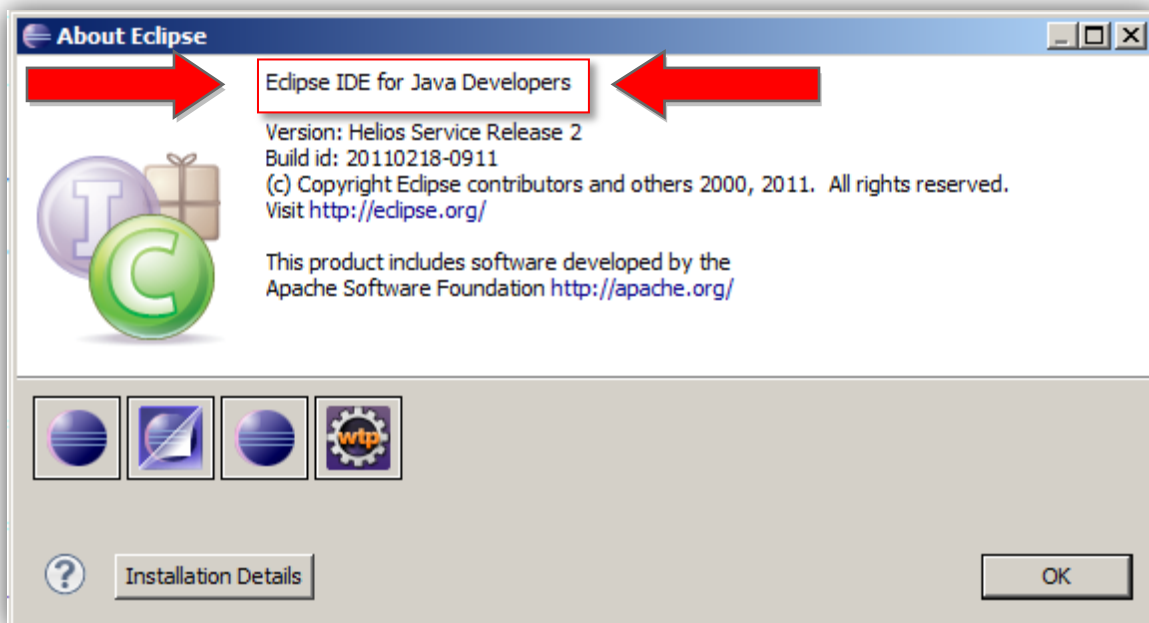
JUnit is dedicated to software developers and testers and is designed for use in testing frameworks. JUnit does come preinstalled on some versions of Eclipse.

Eclipse versions with JUnit preinstalled:

Eclipse JDT

Eclipse IDE for Java EE Development

If you are still not sure which version you have, you can check by opening your download of Eclipse. Once Eclipse opens, navigate to the about screen by clicking **Help > About Eclipse**.



The above instance of Eclipse IDE *does not* have JUnit preinstalled.

There are a few different ways available to get JUnit functionality into your Eclipse instance.

A) Download through Eclipse (Recommended)

In Eclipse IDE, click **Help > Install New Software**. At the work with drop down list, select Helios. This will load a list of software available to download. Look for the Programming languages pack. In this pack is the Eclipse Java Development Tools kit. Downloading this kit will, among other things, give you the functionality to run JUnit test cases.

B) Manually install JUnit Plug-in (Advanced Option)

If you don't want the other tools from step one and want to be sure to only install the specific plug-in, you can manually download and install the plug-in from a source online. There are many different kind of plug-in you can choose from as well.

Once you have the plug-in you desire, place the files in:

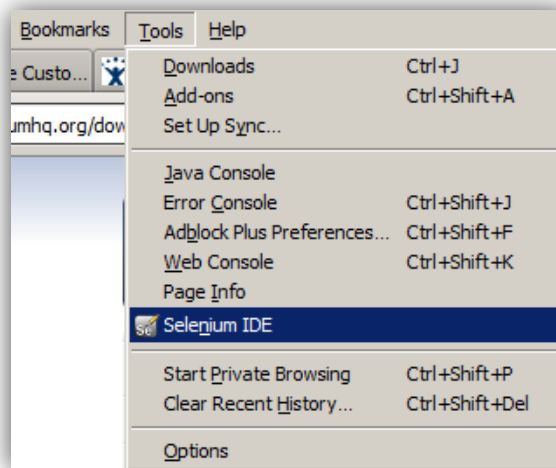
`C:\%ECLIPSE_HOME_DIRECTORY%\plugins\`

Restart Eclipse and you plug-in should be working at that point.

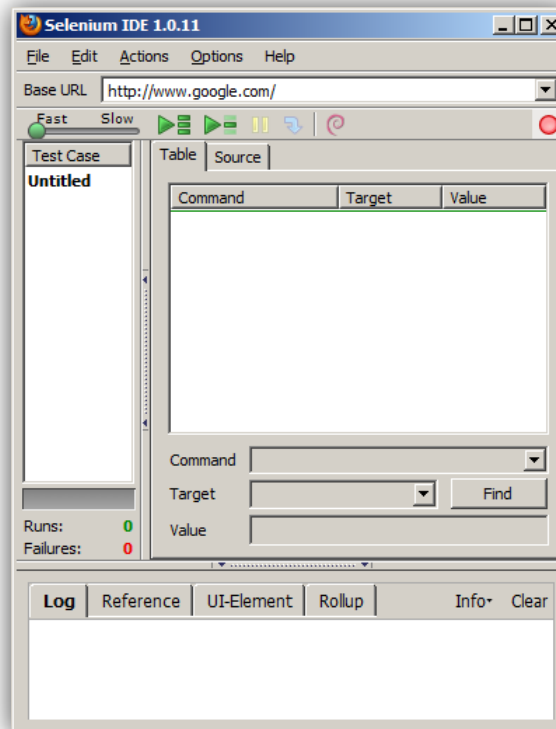
## 3 - Using Selenium (JAVA) with ZBot


### 3.1 Selenium IDE




In Firefox, go to *Tools > Selenium IDE*.

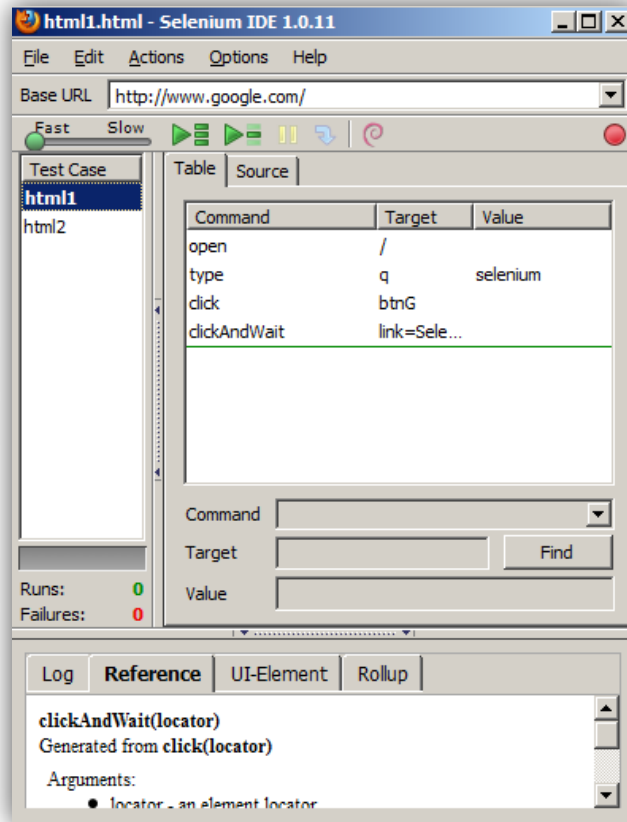


You will be greeted with:

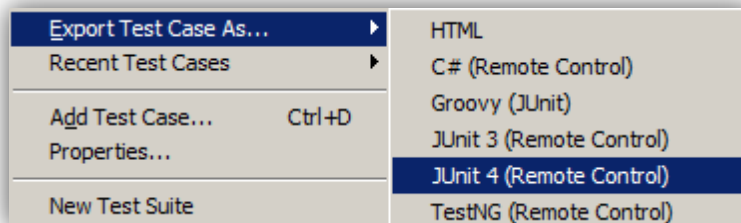


In the top right corner of the Selenium IDE screen, you will see a red button  allows you to start and stop recording. When a recording starts, Selenium will be tracking your actions when using Firefox. Every button you click or field you put text in; Selenium will track and record it in a log.

As an example, click the record button . Go to your favorite search engine, typing something in to search, and then click on any link. Stop the recording after the website loads  and check out the log Selenium makes. All the actions you see on the log are what you did to get to that point. Clicking the play button  to the left of the record button will go through all the actions in the log (quickly). Besides also being able to manually change the log to fit your needs this, in a nutshell, is how to use Selenium IDE.



Once finished creating your test cases, export them by going to **File > Export Test Case As...**. If you created a single selenium folder to hold your files, just create another folder in the root directory called tests.



You will be exporting the Selenium test case in a Java format. For this guide, we will be using the JUnit 4 (Remote Control) export option to create our Java automation script.

### 3.2 Eclipse IDE

Selenium IDE exports files to be used outside of the normal environment. However, the default formats of the test cases straight from Selenium IDE are not completely compatible with what we need to do with them in relation to Zephyr operations (unless you have JUnit framework installed). Thus, some changes need to be made.

Open Eclipse IDE and assign a workspace, if necessary. You should now be on the welcome screen. From here you can go anywhere and do anything within Eclipse IDE.

Find the Package Explorer tab and right click inside of it.

Select **New> Java Project**. This will bring up the New Java Project wizard. Give your project a name, create a package with a name, and then import your java class you exported from Selenium IDE into the package.

This is an example of a simple Google search, exported, and cleaned up, step-by-step:

```
package com.example.tests;  
  
import com.thoughtworks.selenium.*;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import java.util.regex.Pattern;  
  
public class example extends SeleneseTestCase {  
    @Before  
    public void setUp() throws Exception {  
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",  
"http://www.google.com/");  
        selenium.start();  
    }  
  
    @Test  
    public void GoogleSearchExample() throws Exception {  
        selenium.setSpeed("3000");  
        selenium.setTimeout("60000");  
        selenium.open("/");  
        selenium.type("q", "selenium");  
        selenium.click("btnG");  
        selenium.click("link=Selenium web application testing system");  
        selenium.waitForPageToLoad("10000");  
        verifyTrue(selenium.isTextPresent("Get started with  
Selenium!"));  
    }  
}
```

```

    }

    @After
    public void tearDown() throws Exception {
        selenium.stop();
    }
}

```

### Delete:

- `package com.example.tests;`
- `import org.junit.After;`
- `import org.junit.Before;`
- `import org.junit.Test;`
- `import java.util.regex.Pattern;`
- `@Before, @Test, @After`

### Addition:

- `public static Test suite() {  
     return new TestSuite(GoogleSearch3.class);  
 }`
- `public static void main(String args[]) {  
     junit.textui.TestRunner.run(suite());  
 }`

### Revised Version/Sample Code:

```

/*  D SOFTWARE INCORPORATED
 *  Copyright 2007-2011 D Software Incorporated
 *  All Rights Reserved.
 *
 *  NOTICE: D Software permits you to use, modify, and distribute this
file
 *  in accordance with the terms of the license agreement accompanying
it.
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
 */

/*
 * This is a sample of a test case created by Selenium and altered to be
used with Java and run through Zephyr and ZBot
 *
 * Eclipse IDE for Java Developers- Version: Helios Service Release 2.

```

```

Build id: 20110218-0911
* Java- Java JDK 1.6.0_25
*
* Author: Daniel Gannon, Technical Support Analyst, D Software Inc.
*/

//Imported libraries
import com.thoughtworks.selenium.*;
import junit.framework.*;

//Void method that sets up the environment for the test to run
public class GoogleSearch3 extends SeleneseTestCase {
    public void setUp() throws Exception {
        /*
        * localhost is where the Selenium server is located
        * 4444 is the port the server uses
        * *chrome is the browser used in this example. Must start with
a *
        * http://www.google.com/ is the starting URL for the test case
        */
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",
"http://www.google.com/");
        selenium.start();
    }

    //Test case steps
    public void testGoogleSearchExample() throws Exception {
        try{
            //Sets the delay between actions. Some internet services will
require this delay because they will reject automated actions on a
security policy
            selenium.setSpeed("3000");
            //Stops the test 60000 milliseconds after a successful action
that wasn't the last
            selenium.setTimeout("60000");
            selenium.open("/");
            selenium.type("q", "selenium");
            selenium.click("btnG");
            selenium.click("link=Selenium web application testing system");
            selenium.waitForPageToLoad("10000");
            //Text verification to make sure the end website was correct
verifyTrue(selenium.isTextPresent("Get started with
Selenium!"));
            //Catch that takes an exception
        }catch(Exception e){
            //The catch ignores the exception and simply stops the
automation and returns a 1 to the automation tool
            //This is because ZBot can't handle all the resulting text
and logs from a failed test
            //A successful test returns 0(Pass) and a failed test will

```

```


return 1(Fail)
        selenium.stop();
        System.exit(1);
    }
}

//Method Added
public static Test suite() {
    return new TestSuite(GoogleSearchExample.class);
}

//Called when test successfully ends
public void tearDown() throws Exception{
    selenium.stop();
}

//Method Added. Execution starts from here.
public static void main(String args[]) {
    junit.textui.TestRunner.run(suite());
}
}

```

With your code in place, be sure to save your work. Make sure that no error symbols  in your code and then you can debug, compile, and run your code. With this example, debugging is not necessary but is a good practice to understand and use. Also, chances are, if you're using Eclipse IDE, your code is compiling automatically. A tell-tale sign of this is when you're writing code and error/ warning symbols appear without you clicking anything. This means that Eclipse has auto-compile enabled. Lastly, you will want to run your code to be sure that it performs expectedly and outputs correctly before possibly putting it into production. To run your code from the project explorer, **Select Test Case>Right-Click>Run-As>JUnit Test.**

## 3.3 Automation

### 3.3.1 Using the Command Line

Open the command prompt:

### Windows 7/Vista

- Click on the start menu
- In the search bar, input: **CMD**
- Open CMD.EXE

### Windows XP/NT/2000

- Click on the start menu
- Select Run
- Input: **CMD**
- Open CMD.EXE

## 3.3.2 Starting Selenium Server

In order to run Selenium test cases, you will need to be running Selenium Server. The test cases will not run without the server.

The command line structure for a simple server startup batch file with CLASSPATH and path already set will look like this:

```
cd C:\%SELENIUM_HOME_DIRECTORY%\Selenium-remote-control
```

```
Java -jar C:\%SELENIUM_HOME_DIRECTORY%\Selenium-remote-control\selenium-server-standalone-2.0b3.jar -interactive
```

## 3.3.3 Test Case Script

This is an example batch script for the automated test case that will be directed to from the Zephyr test case. ECHO must be turned off in order to stop ZBot from getting anything that it may confuse with exit codes. PATH is set in the script even though it might be set at the system level. Also, because we are using JAVA, we must set the CLASSPATH and JAVA\_HOME. The batch script points to the <filename>.class file created when you compiled the java code in Eclipse IDE. The final line sends the error code back to Zephyr through the ZBot to either pass or fail the test case associated with the batch.

```
@echo off
```

```
SET PATH=C:\Program Files\Java\JDK_1.6.0_<version>\bin;%PATH%  
SET JAVA_HOME=C:\Program Files\Java\JDK_1.6.0_<version>  
SET CLASSPATH=.;%CLASSPATH%
```

```
C:  
  
cd C:\%ECLIPSE_HOME_DIRECTORY%\workspace\GoogleSearchExample  
  
java GoogleSearchExample  
exit %err%
```

You can now run the batch file by itself by double-clicking on it, while the Selenium server is running, and make sure the batch script runs how you expect.


The automation system must both be able to send and receive from the Selenium Server, and have read access to the batch file script for running the test case and the test case <filename>.class java file. Without these, the ZBot will run the batch script but fail because the script could not perform its operations.

### 3.4 Running an Automated Test Case

To use the script file in Zephyr and with ZBot:

#### 1. Create

Create a test case and check the automate checkbox in the test case creation application, and input the path of where the batch file is.



Automated Script

Name: Automation Example

ID: 1

Path: C:\testscript.bat

Priority: P1

History

Created by Test Lead on Jun 14, 2011

#### 2. Assign

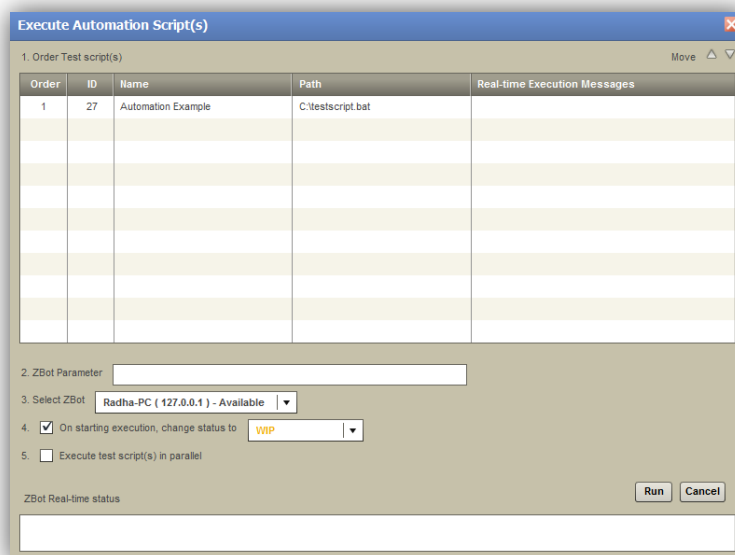
Assign the test case to a tester.

### 3. Execute

The tester can then execute the test case by pressing the 'E' button in test case execution next to a test case marked for automation. Then select the test machine that has ZBot installed. The ZBot will complete the steps and then send back a pass or fail.

ID	Name	Autc	Status	Defect	Notes
27	Automation Example	<b>E</b>	Change Status		

You will be greeted with another screen that will give you the ability to customize your automation run. You can pass parameters to the ZBot, select the ZBot to use, change status of a test case in automation, and finally to run test cases in parallel. When you have your automation run customized to your liking, press **Run**. Status of the run will be logged in the bottom status box. The results of the run will be present to the right of the script in the grid and the test case execution status will update accordingly.



#### Hint

- Pass means the automation encountered no problems
- Fail mean that something stopped the automation from completing
- If in either case, the unexpected happened, it is up to the tester to investigate

For further instructions or clarification, visit: <http://www.zephyrtv.com/>

#### Hint

- Disable pop-up blockers for best viewing of this content

