Selenium for .Net using C# language

I started writing this blog to share my experiences/knowledge on selenium in .Net using C# language. Cuz outside you will get very little information about selenium for .Net.

Saturday, 11 February 2012

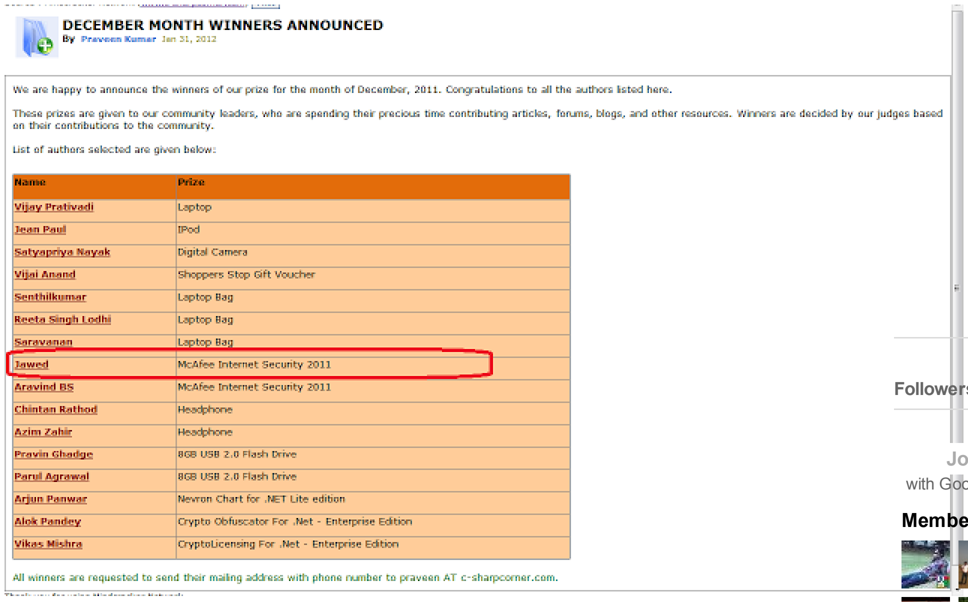
My Name in the list of winners at C-sharp corner for the month of December, 2011.

I was unable to express my happiness when I saw my name in the list of winners.
While posting any article on http://www.c-sharpcorner.com ,I never thought that my name would get listed over there with in 2 months .

But now this became true. All this happen due to all my readers' comments and suggestion which gave me courage to write more and more.
Thanks to all my readers, http://www.c-sharpcorner.com and mentors who encourage me share my knowledge and experience.

These kinds of encouragement provide me a better and nice path to share my knowledge/experience with much bigger audiences.

Lists of winners:
http://www.c-sharpcorner.com/News/1338/december-month-winners-announced.aspx



DECEMBER MONTH WINNERS ANNOUNCED
By Praveen Kumar Jan 31, 2012

We are happy to announce the winners of our prize for the month of December, 2011. Congratulations to all the authors listed here.

These prizes are given to our community leaders, who are spending their precious time contributing articles, forums, blogs, and other resources. Winners are decided by our judges based on their contributions to the community.

List of authors selected are given below:

Name	Prize
Vijay Prativadi	Laptop
Jean Paul	IPod
Satyapriya Nayak	Digital Camera
Vijai Anand	Shoppers Stop Gift Voucher
Senthilkumar	Laptop Bag
Reeta Singh Lodhi	Laptop Bag
Saravanan	Laptop Bag
Jawed	McAfee Internet Security 2011
Aravind BS	McAfee Internet Security 2011
Chintan Rathod	Headphone
Azim Zahir	Headphone
Pravin Ghadge	8GB USB 2.0 Flash Drive
Parul Agrawal	8GB USB 2.0 Flash Drive
Arjun Panwar	Nevron Chart for .NET Lite edition
Alok Pandey	Crypto Obfuscator For .Net - Enterprise Edition
Vikas Mishra	CryptoLicensing For .Net - Enterprise Edition

All winners are requested to send their mailing address with phone number to praveen AT c-sharpcorner.com.

Thanks,
Md. jawed

Posted by MD.JAWED at 09:38 3 comments:

+1 Recommend this on Google Links to this post

Followers

Join this site
with Google Friend Connect

Members (15)

Already a member? Sign in

Blog Archive

- ▼ 2012 (7)
  - ▼ February (3)
    - My Name in the list of winners at C-sharp corner f...
    - Data Driven Testing Using Selenium (webdriver) In ...
    - Get value from Textbox in .Net(C#)
  - ► January (4)

Wednesday, 8 February 2012

Data Driven Testing Using Selenium (webdriver) In C#

Data Driven Testing Using Selenium (webdriver) In C#

SourceCode

In this article, I would like to explain about performing data driven testing in selenium .net.

I have used C# language and for IDE is Visual Studio 2010 ultimate edition to achieve our aim. As we will proceed we will know, steps by steps to Create Data Driven Test method using Selenium in .Net. So for now, let's roll our sleeve and get going.

Before going into details, it would be better to provide synopsis of few things so that even you will get familiar with everything. And as I will proceed you will not feel out of cut from the main theme.

Selenium

Selenium is a portable software testing framework for web applications. Selenium provides a record/playback tool for authoring tests without learning a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including C#, Java, Groovy, Perl, PHP, Python and Ruby. The tests can then be run against most modern web browsers. Selenium deploys on Windows, Linux, and Macintosh platforms.

(Source: http://en.wikipedia.org/wiki/Selenium_(software))

I would refrain myself by going into details about the selenium. As there is tons of documentation and information available on net which already explain about the selenium. You can go through over there.

You can visit selenium official website for the documentation:

http://seleniumhq.org/
Download the selenium for .Net from here:
http://seleniumhq.org/download/
Apart from that you can visit my blog also to get to know about selenium web driver using .Net
http://seleniumdotnet.blogspot.com.

Data Driven Testing

Data driven testing is an action through which set of test inputs and/or output values are read from data files (ODBC source, CSV files, Excel files, DAO objects, ADO objects etc) and are loaded into variables in captured or manually coded script. Data-Driven testing generally means executing a set of steps with multiple sets of data. Selenium does not provide any out-of-the box solution for data driven testing but leaves it up to the user to implement this on his own. That why we here to do out of box ☺

Code:

Prerequisites

- Visual Studio 2010 Premium or Visual Studio 2010 Ultimate.

So now our main coding will start from here:

We will be using Unit testing feature provided by VS2010 and C# code for the same.

1. Select Unit test as project from VS2010
   1.1 open your VS2010
   1.2 Click on Test Tab and select New test

About Me

MD.JAWED

Hello I m Md. Jawed..working as Technical Lead..Having Experience in TFS admin, Build & Release, Wix Packaging, C# Development, Automation and Testing... you can contact me for any Freelancing work on mt emailid at: jawed.ace@gmail.com Thanks!!

View my complete profile

News

Apple Google Microsoft
Selenium Automation tool

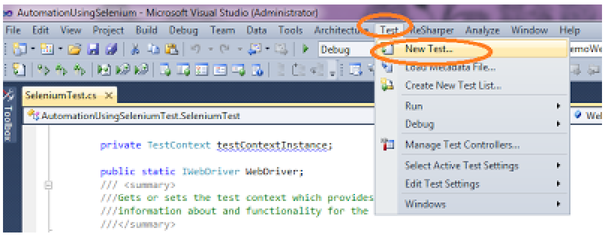
Exclusive: Apple delays HomeKit launch
Fortune
Apple senior vice president of Software Engineering Craig Federighi speaks about the Apple HomeKit app at the Apple Worldwide Developers Conference in San Francisco, Monday, June 2, 2014. Photograph by Jeff Chiu — AP ...
Related Articles »

A Cheap (But Risky) Way to Gold-Plate Your
Wired
For many of us, when the gold Apple Watch Edition was revealed, our inner argonaut screamed, "Yes!" but our bank accounts said, "Um, no." The fact is, few people have 10 grand to drop on a gadget. But where there's a will, there's a way, and the ...

The Apple Watch is no harder to steal than a
The Verge
9to5Mac is probably raising some concern among Apple Watch users today, claiming that an a "bug" allows thieves to easily factory reset Apple's smartwatch without knowing your personal code. And everything about that sentence is true except for one ...
Related Articles »

Apple, Spotify, and the Battle over Freemium
blogs.hbr.org (blog)
Just when all the hype about the release of Apple Watch was starting to fade away, Apple made headlines again. This time, it's due to reports that the company is pushing record labels to force streaming services like Spotify to abandon their free tiers.
Related Articles »

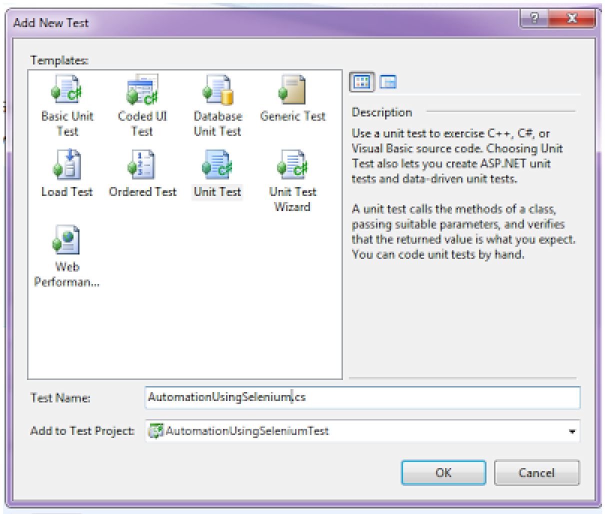
powered by Google™



```
private TestContext testContextInstance;

public static IWebDriver WebDriver;
/// <summary>
///Gets or sets the test context which provides
///information about and functionality for the
///</summary>
```

Figure 1: Select New test

1.3 Click on Unit test from the New open window



Use a unit test to exercise C++, C#, or Visual Basic source code. Choosing Unit Test also lets you create ASP.NET unit tests and data-driven unit tests.

A unit test calls the methods of a class, passing suitable parameters, and verifies that the returned value is what you expect. You can code unit tests by hand.

Test Name: AutomationUsingSelenium.cs

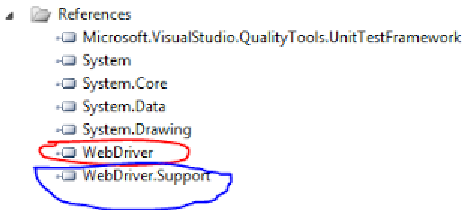
Add to Test Project: AutomationUsingSeleniumTest

Figure2. Select Unit Test from VS2010

1.4 Rename the File to SeleniumTest.cs

2. Add Selenium web driver references

2.1 Add below dlls through add references



```
References
    Microsoft.VisualStudio.QualityTools.UnitTestFramework
    System
    System.Core
    System.Data
    System.Drawing
    WebDriver
    WebDriver.Support
```

Figure3. Add Selenium DLL in the project.

3. Open you selenium.cs file and add below Name space

```
using OpenQA.Selenium;
using OpenQA.Selenium.IE;
//add this name space to access WebDriverWait
using OpenQA.Selenium.Support.UI;
```

Never try to add this forcefully until unless you are not getting into your VS intelligence.

Test Initialization:

```
public static IWebDriver WebDriver;

// Use TestInitialize to run code before running each test
        [TestInitialize()]
        public void MyTestInitialize()
        {
            var capabilitiesInternet = new
OpenQA.Selenium.Remote.DesiredCapabilities();
```

```
            capabilitiesInternet.SetCapability("ignoreProtectedModeSettings",
true);
            WebDriver = new InternetExplorerDriver(capabilitiesInternet);

        }
```

Test Cleanup:

```
// Use TestCleanup to run code after each test has run
        [TestCleanup()]
        public void MyTestCleanup()
        {

            WebDriver.Quit();

        }
```

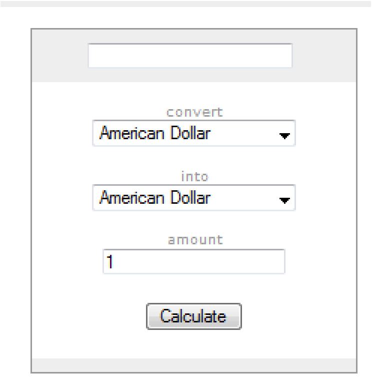
Before implementing the test method for data driven we will write the selenium Test method without Data Driven. Then we will go for Data Driven Testing, by going with this scenario you will clearly understand the difference between simple selenium test method without data driven and selenium test method with Data Driven.

Selenium Test Method without Data Driven

To write Automation code using selenium to automate our target website we will use any web application. Here I will use http://www.x-rates.com/calculator.html web application for our AUT (Application under Test) purpose. The function of this AUT web application is to compare e currency value from one country’s currency to other country’s currency.

NOTE: The AUT website is just for demo purpose. The currency rate would not be fixed. So might be the test cases pass for me would failed for you or vise-versa.

The web page is as below:



convert: American Dollar

into: American Dollar

amount: 1

Figure 4. Application Under test

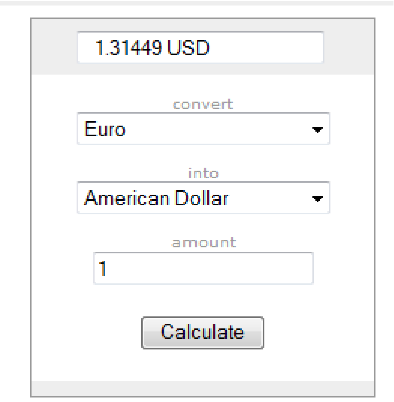
Now we will write our Selenium Test method to perform below operation:

1. Select Value in to first dropdown box(convert)
2. Select value into second dropdown box(into)
3. Click on Calculate button to get the currency value w.r.t first country’s currency selected.
4. Read the value from upper textbox/input box. This will be our actual result.
5. Compare the actual value from expected value.

```
        [TestMethod]
         public void TestCurrencyConvertorWithoutDDT()
         {
             //Read your first country currency name
             var convertVal = "American Dollar";
             //Read your second contry currency
             var inToVal = "Indian Rupee";
             //Read Expected value from data source
             var expectedValue = "49.7597 INR";
             //Goto the Target website
             WebDriver.Navigate().GoToUrl("http://www.x-
                               rates.com/calculator.html");
             var setValueConvert = WebDriver.FindElement(By.Name("from"));
             var setValueInto = WebDriver.FindElement(By.Name("to"));
             var calculateButton = WebDriver.FindElement(By.Name("Calculate"));
             var outPutvalue = WebDriver.FindElement(By.Name("outV"));
```

```
            var selectConvertItem = new SelectElement(setValueConvert);
            var selectIntoItem = new SelectElement(setValueInto);

            selectConvertItem.SelectByText(convertVal);
            selectIntoItem.SelectByText(inToVal);
            calculateButton.Click();
            var currencyValue = outPutvalue.GetAttribute("value");

            Thread.Sleep(900);//Not a good practise to use Sleep
            //Get the screen shot of the web page and save it on local disk
            SaveScreenShot(WebDriver.Title);
            Assert.AreEqual(expectedValue, currencyValue.Trim());

        }
```

Target Web page shown as below:



1.31449 USD

convert: Euro

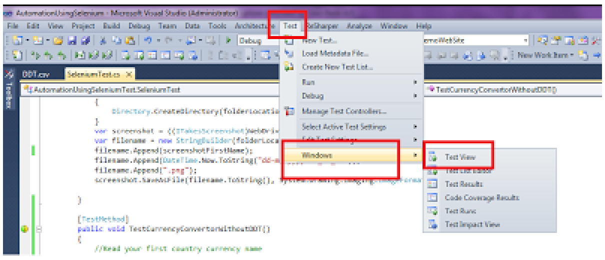
into: American Dollar

amount: 1

Figure 5: Target web page with all the controls

Run the above Test method to see the result:

1. On the **Test** menu, click **Windows** and the select **Test View**.

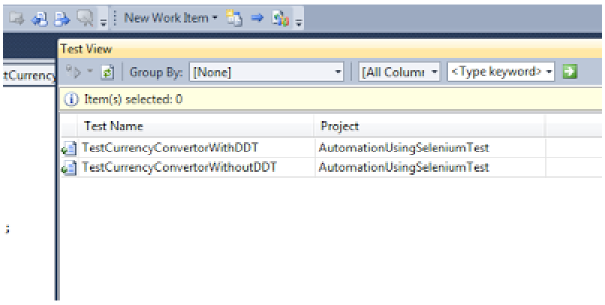


```
{
    Directory.CreateDirectory(folderLocatio
}
var screenshot = ((ITakesScreenshot)WebDriv
var filename = new StringBuilder(folderLoca
filename.Append(screenshotFirstName);
filename.Append(DateTime.Now.ToString("dd-m
filename.Append(".png");
screenshot.SaveAsFile(filename.ToString(), System.Drawing.Imaging.ImageForma

}

[TestMethod]
public void TestCurrencyConvertorWithoutDDT()
{
    //Read your first country currency name
```

AutomationUsingSelenium - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Data Tools Architecture Test ReSharper Analyze Window Help
Debug
New Work Item
DDT.csv
SeleniumTest.cs
AutomationUsingSeleniumTest.SeleniumTest
TestCurrencyConvertorWithoutDDT()
New Test...
Load Metadata File...
Create New Test List...
Run
Debug
Manage Test Controllers...
Select Active Test Settings
Edit Test Settings
Windows
Test View
Test List Editor
Test Results
Code Coverage Results
Test Runs
Test Impact View

Figure 6. Select Test View

The Test View window is displayed.



Item(s) selected: 0

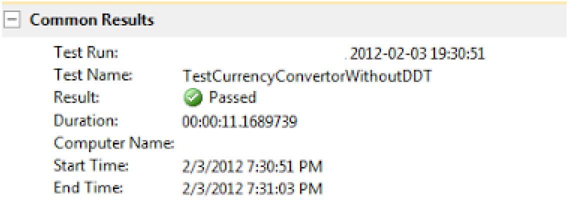
Test Name	Project
TestCurrencyConvertorWithDDT	AutomationUsingSeleniumTest
TestCurrencyConvertorWithoutDDT	AutomationUsingSeleniumTest

Figure 7. Test View Window

2. Right-click **TestCurrencyConvertorWithoutDDT** and click **Run Selection**.

If the **Test Results** window is not already open, it opens now. The **TestCurrencyConvertorWithoutDDT** test runs.
In the **Result** column in the **Test Results** window, test status is displayed as **Running** while the test is running.

3. In the **Test Results** window, right-click the row that represents the test and then click **View Test Results Details**.



Common Results

Test Run:	2012-02-03 19:30:51
Test Name:	TestCurrencyConvertorWithoutDDT
Result:	✓ Passed
Duration:	00:00:11.1689739
Computer Name:	
Start Time:	2/3/2012 7:30:51 PM
End Time:	2/3/2012 7:31:03 PM

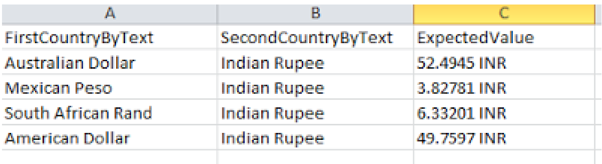
Figure 8. Test Results Windows

Test Method as Data Driven

As already you have seen in above Selenium test method we ran our method for only one set on values. Now in Data Driven we will run the same method with multiple sets of inputs value.

To read the inputs value against which we are going to run our Test method, we have stored our inputs values into some data Source. For that we will use CSV file, but you can use other data source also like excel Sheet else.
In this CSV file we will store our expected result also so that we can compare our actual output with this value.

Our csv would look something like this (give the name to this csv file as DDT.csv):

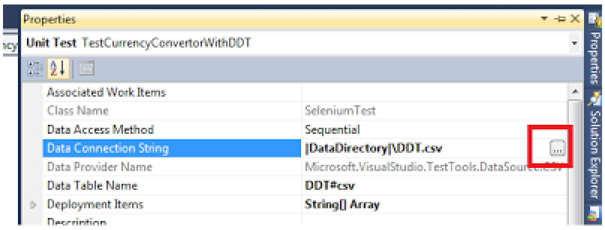


A	B	C
FirstCountryByText	SecondCountryByText	ExpectedValue
Australian Dollar	Indian Rupee	52.4945 INR
Mexican Peso	Indian Rupee	3.82781 INR
South African Rand	Indian Rupee	6.33201 INR
American Dollar	Indian Rupee	49.7597 INR

Figure 9. CSV file as Data Source

Now we will connect the above Test method to our data Source [here it is DDT.CSV]

1. Open the solution that contains the test method for which you want to use a data source.
2. On the **Test** menu, point to **Windows**, and then click **Test View**.
3. In the **Test View** window, right-click the unit test for which you want to use a data source and then click **Properties**.

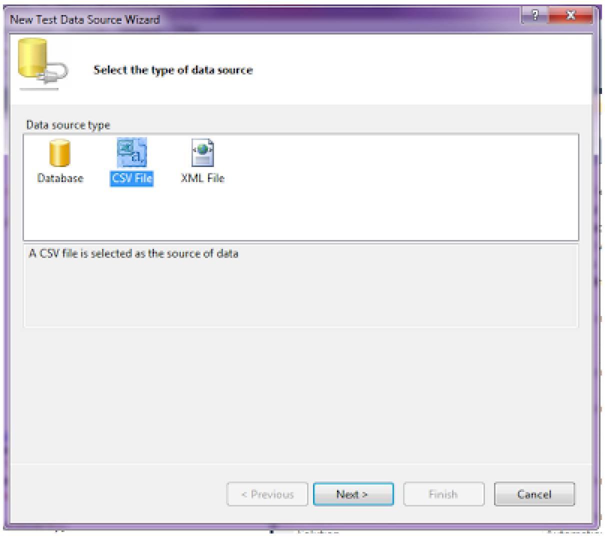


Unit Test TestCurrencyConvertorWithDDT

Associated Work Items	
Class Name	SeleniumTest
Data Access Method	Sequential
Data Connection String	|DataDirectory|\DDT.csv
Data Provider Name	Microsoft.VisualStudio.TestTools.DataSource.CSV
Data Table Name	DDT#csv
Deployment Items	String[] Array
Description	

Figure 10. Properties of Selenium Test Method

4. In the **Properties** window click **Data Connection String** and then click the ellipses(…).



New Test Data Source Wizard
Select the type of data source
Data source type
Database
CSV File
XML File
A CSV file is selected as the source of data
< Previous
Next >
Finish
Cancel

Figure 11. **Test Data Source Wizard**

5. Follow the instructions in the in the **New Test Data Source Wizard** to create the data connection.
   A connection string is added to your unit test after the first bracket of the [TestMethod()] element.
   It would be something like this:

```
[DeploymentItem("AutomationUsingSeleniumTest\\DDT.csv"),
DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",
"|DataDirectory|\\DDT.csv", "DDT#csv", DataAccessMethod.Sequential)]
```

6. **Now we will assign variables to values from your data source**
   6.1 Open the Test file that contains the test method for which you want to use a data source and locate the variables in the test method.

   6.2 For each variable that you want to come from the data source, use the syntax TestContext.DataRow["NameOfColumn"].

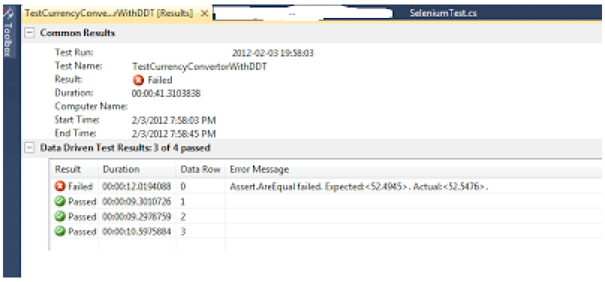
   The piece of code would be as below:

```
 //Read your first country currency name
var convertVal = TestContext.DataRow["FirstCountryByText"].ToString();
//Read your second contry currency
var inToVal = TestContext.DataRow["SecondCountryByText"].ToString();
//Read Expected value from data source
var expectedValue = TestContext.DataRow["ExpectedValue"].ToString();
```

The whole Code of the Test Method would be as below:

```
[DeploymentItem("AutomationUsingSeleniumTest\\DDT.csv"),
DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",
"|DataDirectory|\\DDT.csv", "DDT#csv", DataAccessMethod.Sequential)]
        [TestMethod]
        public void TestCurrencyConvertorWithDDT()
        {

           //Read your first country currency name
            var convertVal = TestContext.DataRow["FirstCountryByText"].ToString();
           //Read your second contry currency
            var inToVal = TestContext.DataRow["SecondCountryByText"].ToString();
           //Read Expected value from data source
            var expectedValue = TestContext.DataRow["ExpectedValue"].ToString();
           //Goto the Target website
            WebDriver.Navigate().GoToUrl("http://www.x-rates.com/calculator.html");
            var setValueConvert = WebDriver.FindElement(By.Name("from"));
            var setValueInto = WebDriver.FindElement(By.Name("to"));
            var calculateButton = WebDriver.FindElement(By.Name("Calculate"));
            var outPutvalue = WebDriver.FindElement(By.Name("outV"));

            var selectConvertItem=new SelectElement(setValueConvert);
            var selectIntoItem = new SelectElement(setValueInto);

            selectConvertItem.SelectByText(convertVal);
            selectIntoItem.SelectByText(inToVal);
            calculateButton.Click();
            var currencyValue =outPutvalue.GetAttribute("value") ;

            Thread.Sleep(900);//Not a good practise to use Sleep
           //Get the screen shot of the web page and save it on local disk
           SaveScreenShot(WebDriver.Title);
            Assert.AreEqual(expectedValue,currencyValue.Trim());

        }
```

7. Run the Test Method (already I explained this already) and Result would be something like below:



TestCurrencyConve...rWithDDT [Results] SeleniumTest.cs

Common Results

Test Run: 2012-02-03 19:58:03
Test Name: TestCurrencyConvertorWithDDT
Result: Failed
Duration: 00:00:41.3103838
Computer Name:
Start Time: 2/3/2012 7:58:03 PM
End Time: 2/3/2012 7:58:45 PM

Data Driven Test Results: 3 of 4 passed

Result	Duration	Data Row	Error Message
Failed	00:00:12.0194088	0	Assert.AreEqual failed. Expected:<52.4945>. Actual:<52.5476>.
Passed	00:00:09.3010726	1	
Passed	00:00:09.2978759	2	
Passed	00:00:10.5975884	3	

Figure 12. Data Driven Test Results

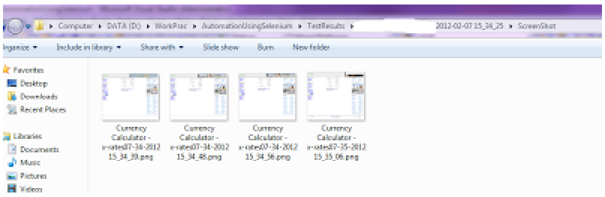
So you can clearly see that the Test method had ran for 4 different set of value. And it fetched the input values from our specified data source that is DDT.csv.

Download the code to get the whole piece of code. I have used a method that is **SaveScreenShot(WebDriver.Title).** You can get the code details along with the uploaded code. Actually this method would take the screen shot you your target web page and will store that on Test Result directory.

Save Screen Shot Method

```
/// <summary>
        /// This will Take the screen shot of the webpage and will save it at
particular location
        /// </summary>
        /// <param name="screenshotFirstName"></param>
        private static void SaveScreenShot(string screenshotFirstName)
        {
            var folderLocation =
Environment.CurrentDirectory.Replace("Out","")+"\\ScreenShot\\";
            if (!Directory.Exists(folderLocation))
            {
                Directory.CreateDirectory(folderLocation);
            }
            var screenshot = ((ITakesScreenshot)WebDriver).GetScreenshot();
            var filename = new StringBuilder(folderLocation);
            filename.Append(screenshotFirstName);
            filename.Append(DateTime.Now.ToString("dd-mm-yyyy HH_mm_ss"));
            filename.Append(".png");
            screenshot.SaveAsFile(filename.ToString(),
System.Drawing.Imaging.ImageFormat.Png);

        }
```

The Screen shot of your Target web page would get saved as below:



Computer ▸ DATA (D:) ▸ WorkProc ▸ AutomationUsingSelenium ▸ TestResults ▸ 2012-02-07 15_34_25 ▸ ScreenShot
Organize ▾ Include in library ▾ Share with ▾ Slide show Burn New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Currency Calculator - x-rates07-34-2012 15_34_39.png
Currency Calculator - x-rates07-34-2012 15_34_48.png
Currency Calculator - x-rates07-34-2012 15_34_56.png
Currency Calculator - x-rates07-35-2012 15_35_06.png

Figure 13. Screen Shot saved at Test Result folder.

Feel free to provide your feedback and comments!!
For any issue/question/suggestion in selenium for C# you refer this blog
http://seleniumdotnet.blogspot.com

Thanks,
MD. Jawed (jawed.ace@gmail.com)

Posted by MD.JAWED at 12:49 3 comments:

g+1 +3 Recommend this on Google Links to this post

Labels: C#, Data Driven, Selenium, VS2010

Home Older Posts

Subscribe to: Posts (Atom)

Total Pageviews

9 3 2 6 2