

Visual Studio 2010 Coded UI Test User Guide

Written By: [Eran Ruso](#)

Table of Content

1. INTRODUCTION	3
2. CREATE A SIMPLE CODED UI TEST	4
3. GET A CODED UI TEST CREATED IN MANUAL TEST USING THE MTLM TOOL.....	13
4. ADD VALIDATION TO YOUR CODED UI TESTS	15
5. SUMMARY	19

1. Introduction

In Visual Studio 2010 Microsoft has added the Coded UI Test.

Back to history:

In Visual Studio 2005 Microsoft presented the Tester Edition, it contained the following parts:

1. Web Test Automation Test
2. Performance Test
3. Unit Test
4. Something that should have been a Manual Test, it was word or text documents contained the Test Cases. Practically no one used it.

In Visual Studio 2008 tester edition no significant changes has been made.

In Visual Studio 2010 Microsoft added more parts to the Tester Edition and created a new tool for the Tester called Microsoft Test and Lab Management (MTLM).

I will not get into all the new functionality added to Team System 2010 but I will list the main parts:

1. Web
2. Web Test Automation Test
3. Performance Test
4. Unit Test
5. Test Cases management
6. Manual tests
7. **Coded UI Tests**
8. Lab Management

In the following 3 posts of the user guide I will focus on **the main tasks using the Coded UI Test, best practices** with the tool and with automation tests in general.

2. Create a Simple Coded UI Test

On this Chapter I will show how to create a simple Coded UI Test.

The example I will use in this guide created by Microsoft and is called: “IBuySpay”

The example is a web application and we will use its GUI to demonstrate the Coded UI Test.

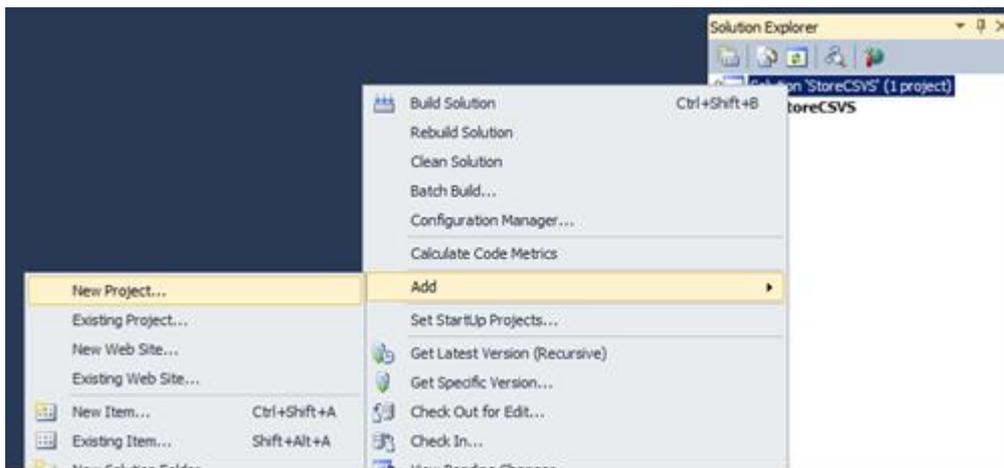
A few guide lines before we start:

1. To create a test you need to have a Test Project
2. Base every Automated Test on a predefined Test Case
3. Every step in the Test Case will be a Method in the Coded UI Test

Note: In this example I will not show the Test Case.

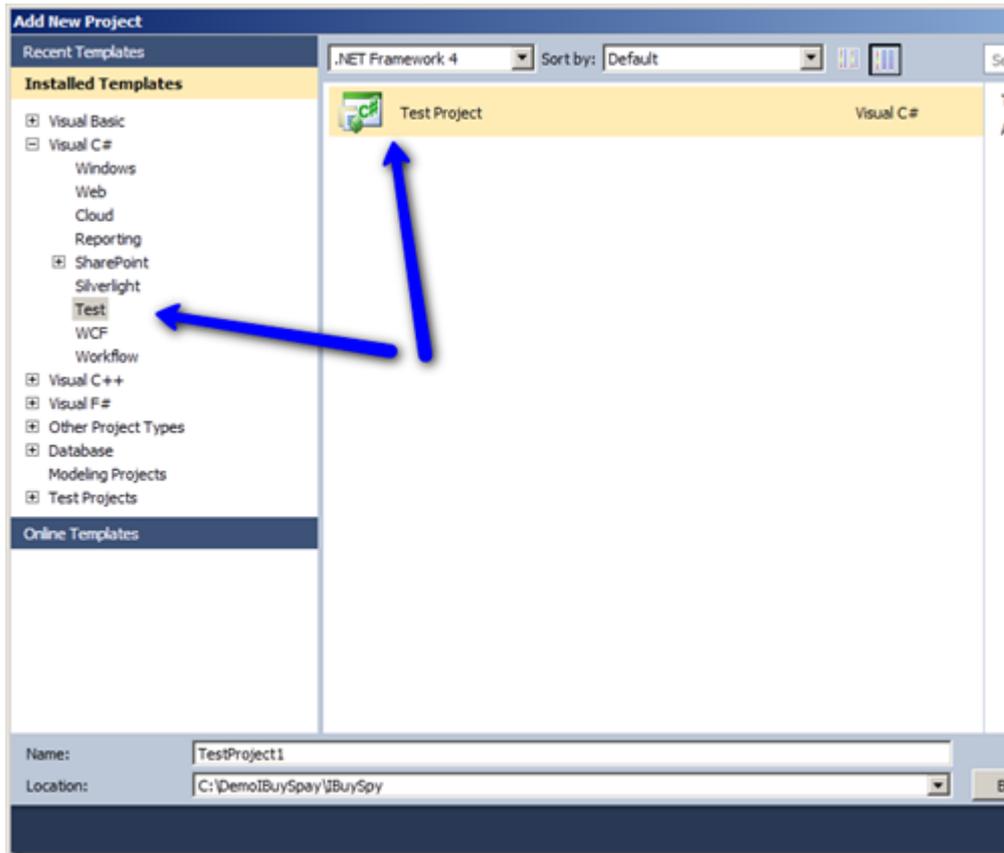
Creating a Test Project

Step 1: Right click the Solution choose: Add → New Project...



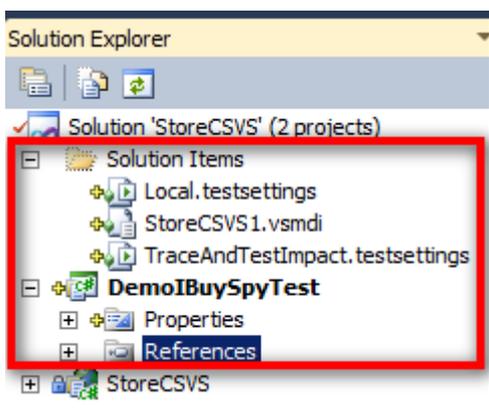
Step 2: Choose Test from the Menu, in my example it will be a C# Test Project

Step 3: Choose the location and the name of the test project and choose Create



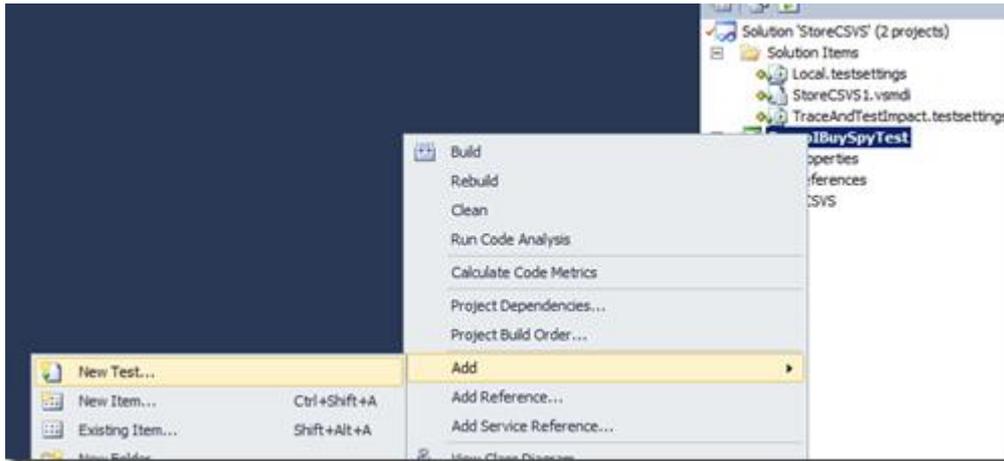
The result is the test project (DemoIBuySpyTest) and 3 more solution items that are added in case this is the first test project.

These 3 files are out of the scope of this guide.



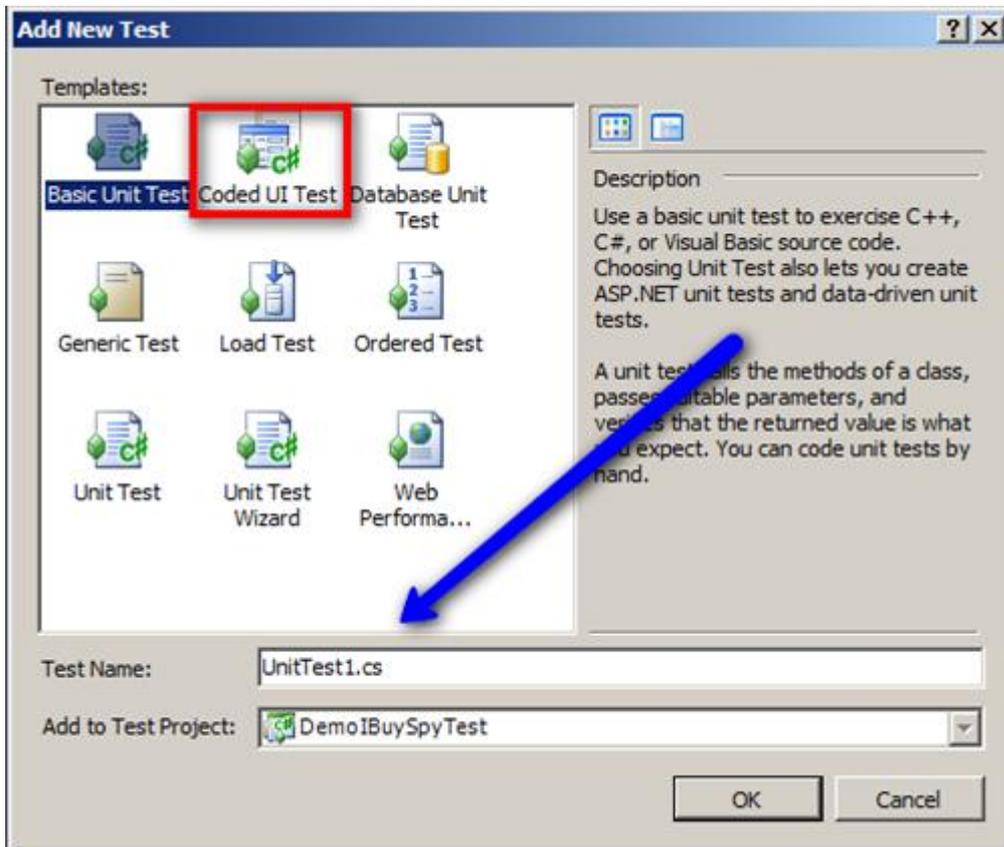
Creating a Coded UI Test

Step 1: Right click the Test Project choose Add → New Test...



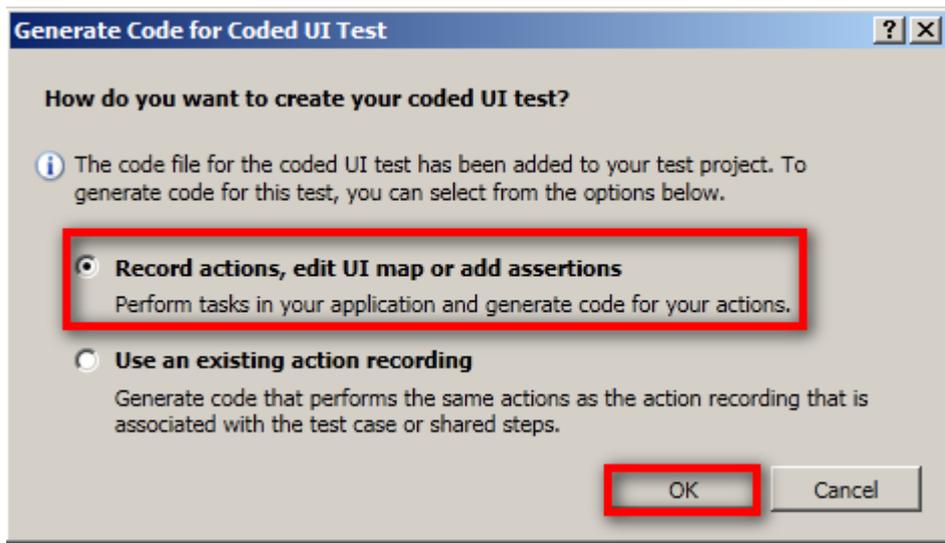
Step 2: Choose Coded UI Test

Step 3: Type a Test name and click OK



Step 4: In the window that pops up choose the first option: “Record actions, edit UI map or add assertions”

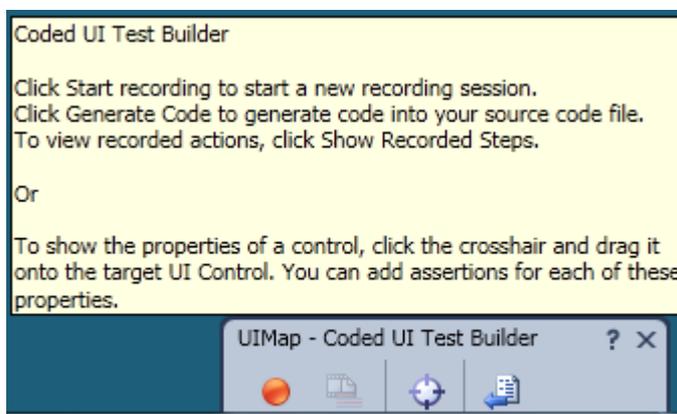
Step 5: Click OK



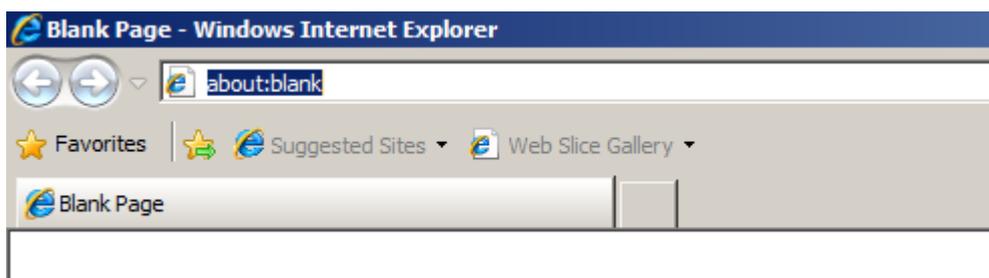
The Recorder launches.

The Recorder has 4 buttons (From left to right):

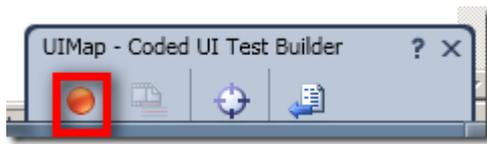
1. **Record** – Will record actions
2. **Show Recorder Steps** – Shows what was recorder and lets the user edit the steps
3. **Add Assertion** – Lets the user choose objects and to get there “Passport”, it also lets the user add validation to objects
4. **Generate Code** – Lets the user create code out of the recorder actions



Step 6: launch the Internet Explorer Browser



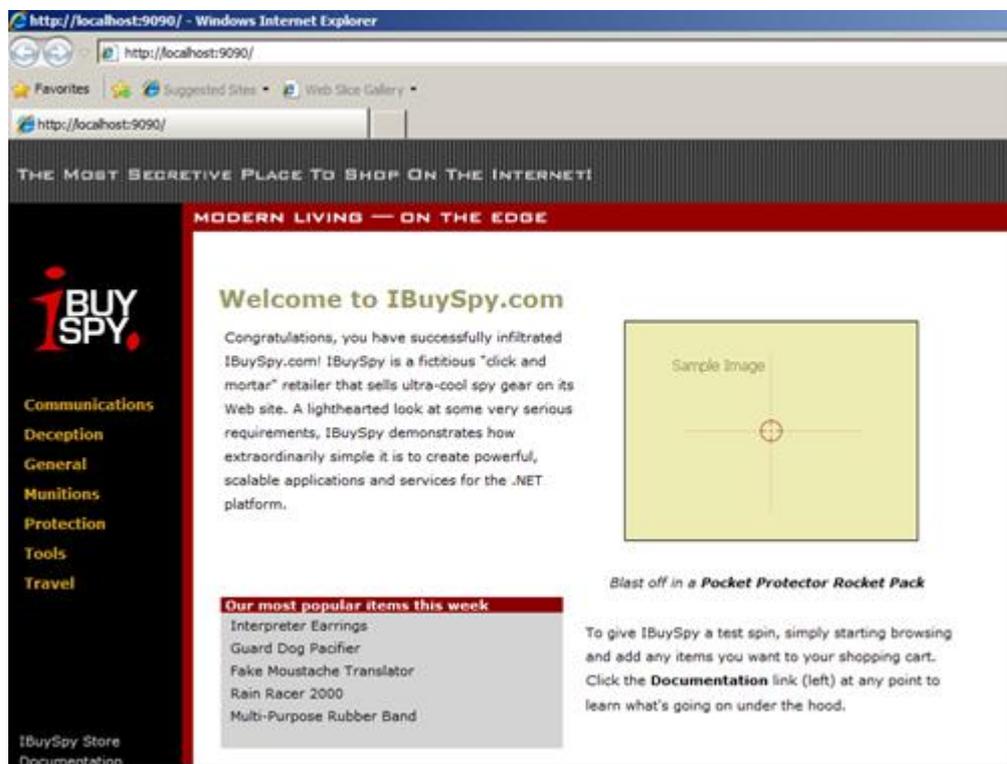
Step 7: Click Record



The icon will be changed to Pause



Step 8: Type the URL of the tested site



Following the guide lines we need to create a method for each step in the Test Case

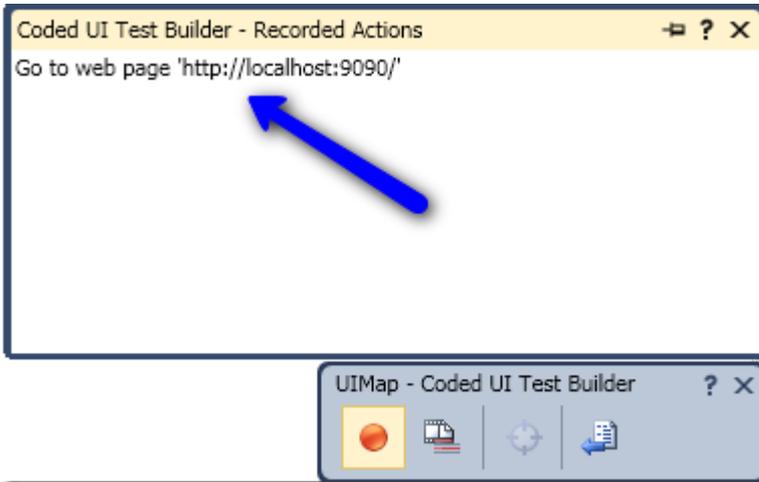
Step 9: Click the Pause button



Step 10: Click the “Show Recorder Steps” in order to see and edit the recorder steps



You can see that there is only one recorded steps, in order to edit the recorded steps (If needed) choose the step and click the Delete button in your keyboard.

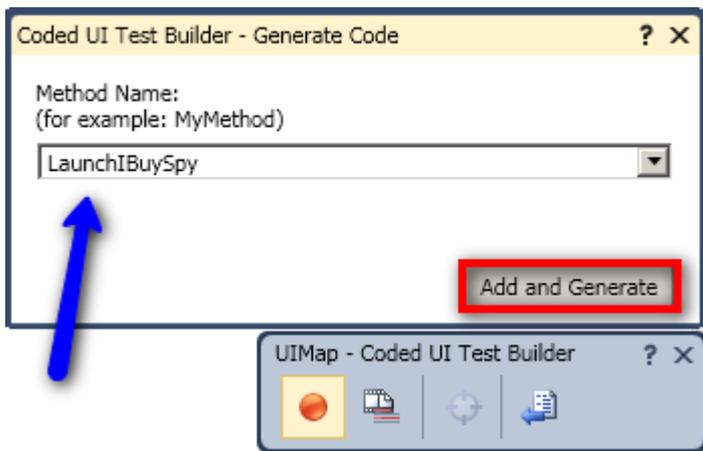


Step 11: Click the “Generate Code”



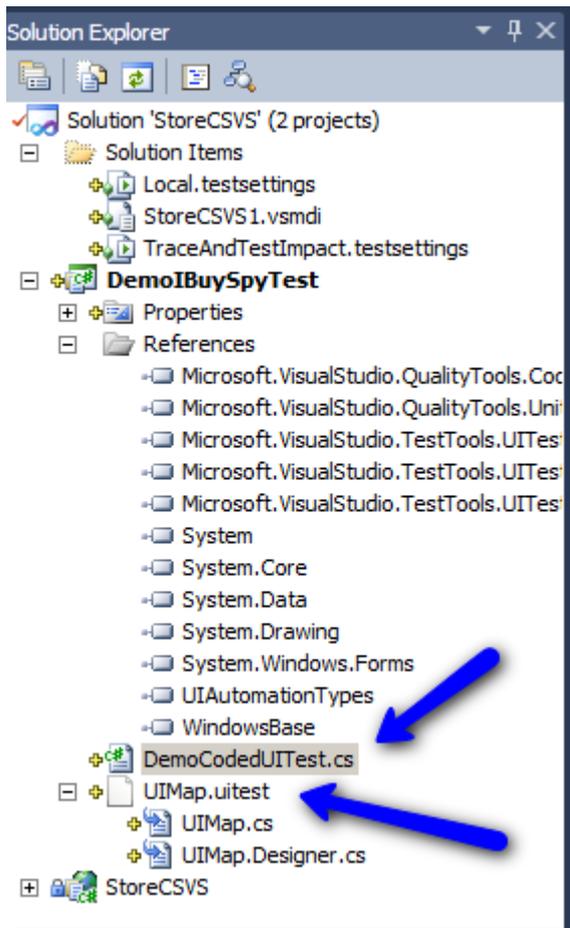
Step 12: Type the name of the method

Step 13: Click the “Add and Generate” button



The result is the code generated in the “DemoCodedUITest.cs” TestMethod and new files created, these files have the UI objects mapping.

You can also see the Method we have created.



```

namespace DemoIBuySpyTest
{
    /// <summary>
    /// Summary description for DemoCodedUITest
    /// </summary>
    [CodedUITest]
    public class DemoCodedUITest
    {
        public DemoCodedUITest()
        {
        }

        [TestMethod]
        public void CodedUITestMethod1()
        {
            // To generate code for this test, select "Generate Code for Coded
            // For more information on generated code, see http://go.microsoft.com/fwlink/?LinkId=105466
            this.UIMap.LaunchIBuySpy();
        }

        Additional test attributes

        /// <summary>
        /// Gets or sets the test context which provides
        /// information about and functionality for the current test run.
        /// </summary>
        public TestContext TestContext
        {
            get
            {
                return testContextInstance;
            }
        }
    }
}

```

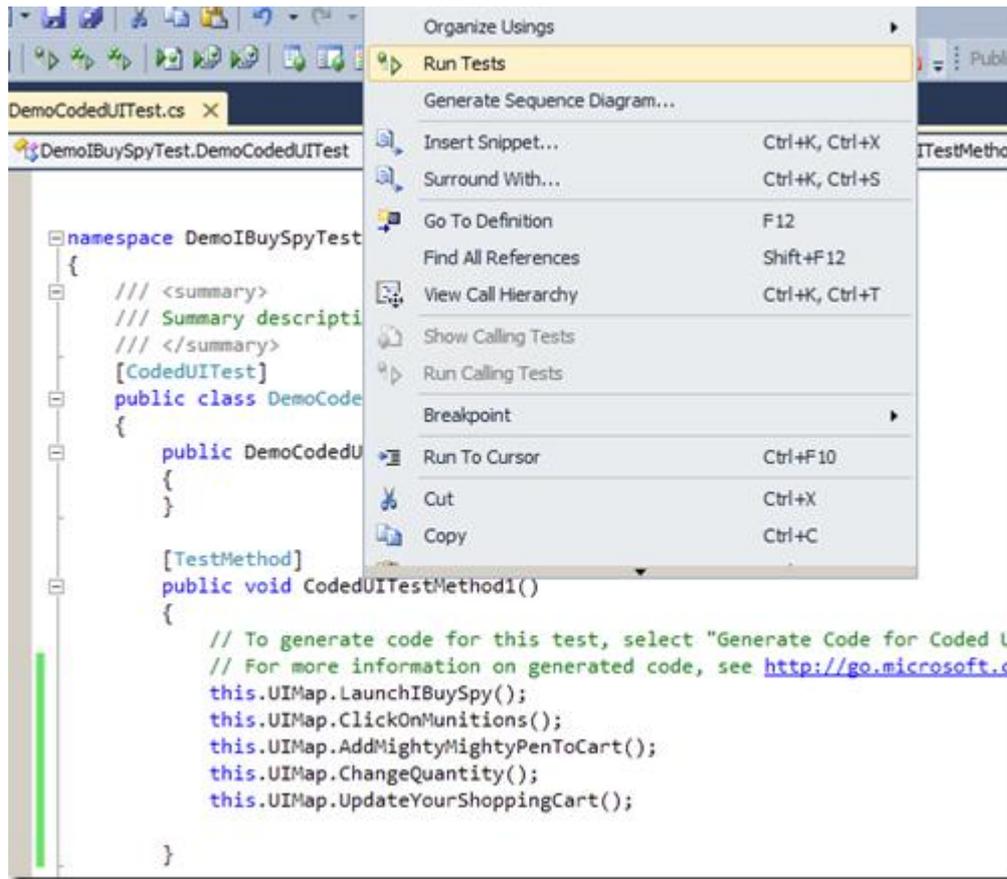
I have created a few more methods that represent a few more steps.

```

[TestMethod]
public void CodedUITestMethod1()
{
    // To generate code for this test, select "G
    // For more information on generated code, s
    this.UIMap.LaunchIBuySpy();
    this.UIMap.ClickOnMunitions();
    this.UIMap.AddMightyMightyPenToCart();
    this.UIMap.ChangeQuantity();
    this.UIMap.UpdateYourShoppingCart();
}

```

In order to run the test you can right click the Coded UI Test name and choose “Run Tests”.



We have created a simple Coded UI Test, next on this guide we will see how to reuse a prerecorded Coded UI Test in the MTLM tool.

3. Get a Coded UI Test Created In Manual Test Using the MTLM Tool

On this Chapter I will show how to create a Coded UI Test based on prerecorded Coded UI Test.

In the Microsoft Test and Lab Management (MTLM) tool you run Test Plans with the Test Runner, **In case that the “Action Log and Action Recording” collector is turned on the Test Runner will create a Coded UI Test in the background while running the Manual Test.**

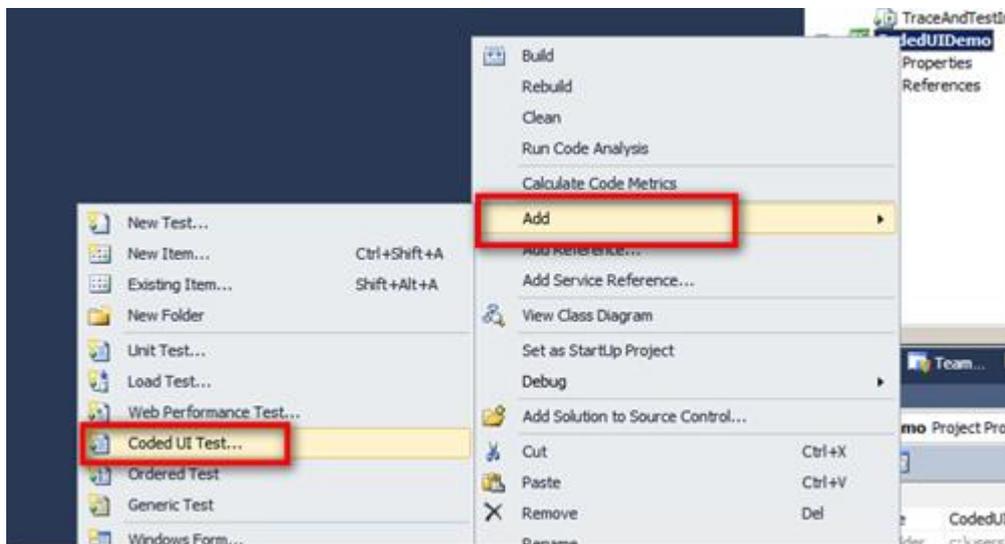
The recorded Coded UI Test is very simple and basic test that **will not answer the Automation Test Effort requirements** but it’s a start.

The Tester that is assigned on creating the Automated Tests can have a very good starting point using these prerecorded tests.

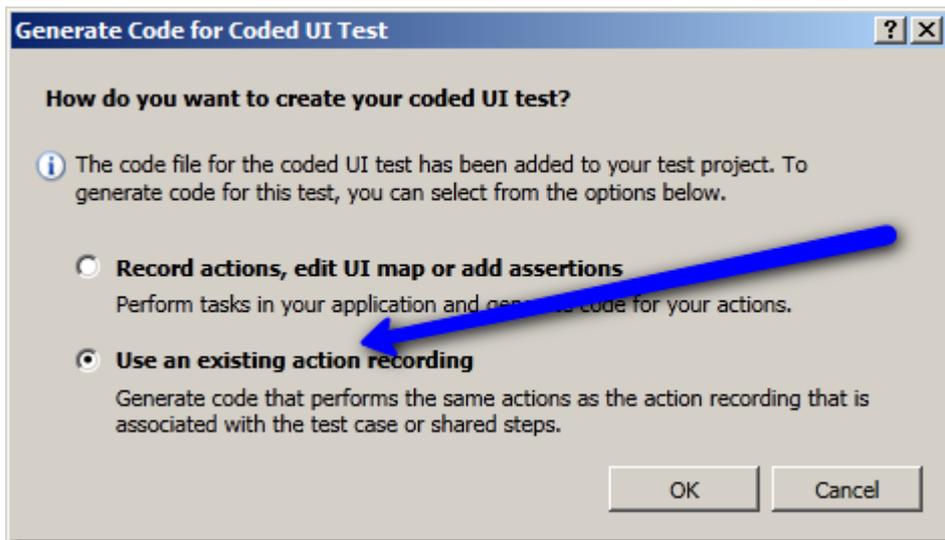
How to create a Coded UI test out of an existing recording?

Step 1: Create or open a Test Project

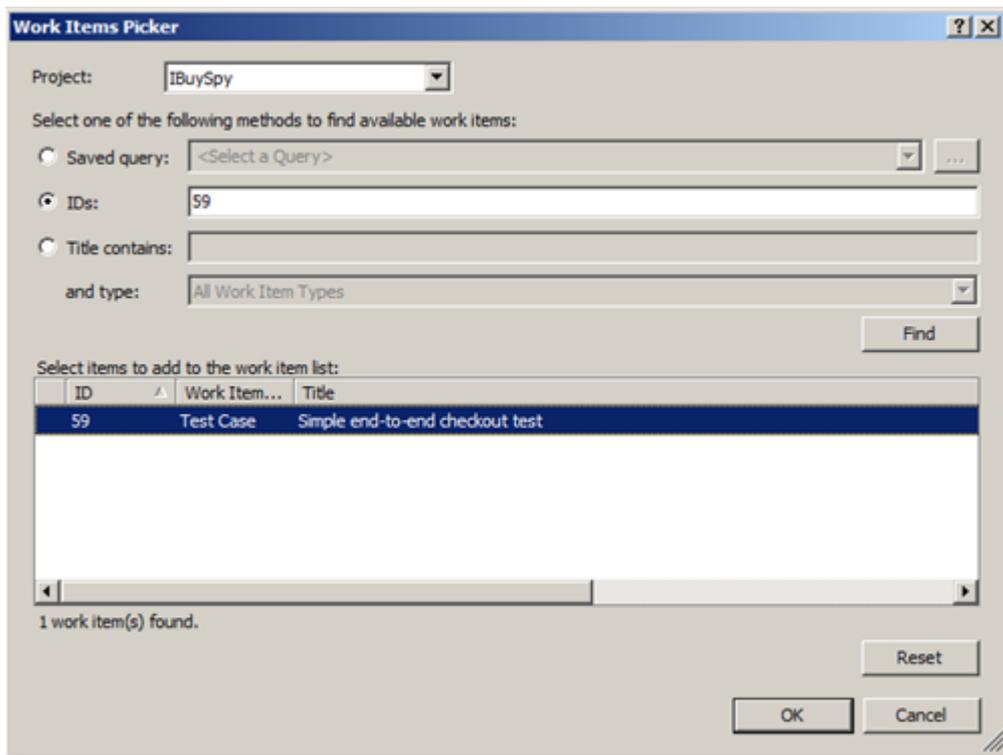
Step 2: Add a new Coded UI Test



Step 3: In the window that Pops Up choose: “Use an existing action recording” radio button and click OK.

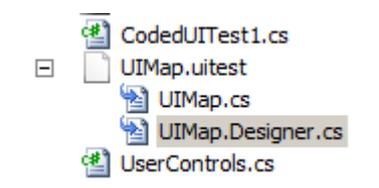


Step 4: Find the Test Case that contains the Coded UI Recording in the window that pops up and click OK.



Finally you will get a code for Coded UI based on the Recording, now all you have to do is modify the code to suit your needs.

Note: The files in the screenshot are the files that will be generated.



4. Add Validation to Your Coded UI Tests

On this Chapter I will show how to add validation rules to the Coded UI Test that we have previously recorded.

This Chapter is based on the example made in the second Chapter of this guide: “Create A Simple Coded UI Test”

Now that we have the code for the Coded UI Test we can add validation rules.

The validation rule I like to add will be placed after the last method of the test “this.UIMap.UpdateYourShoppingCart()” and I will call it: “CheckSubtotal”

Follow these steps to add this validation:

Step 1: Place the mouse cursor after the last method this.UIMap.UpdateYourShoppingCart() in the test.

Step 2: Right click choose “Generate Code for Coded UI Test” -> “Use Coded UI Test Builder...”

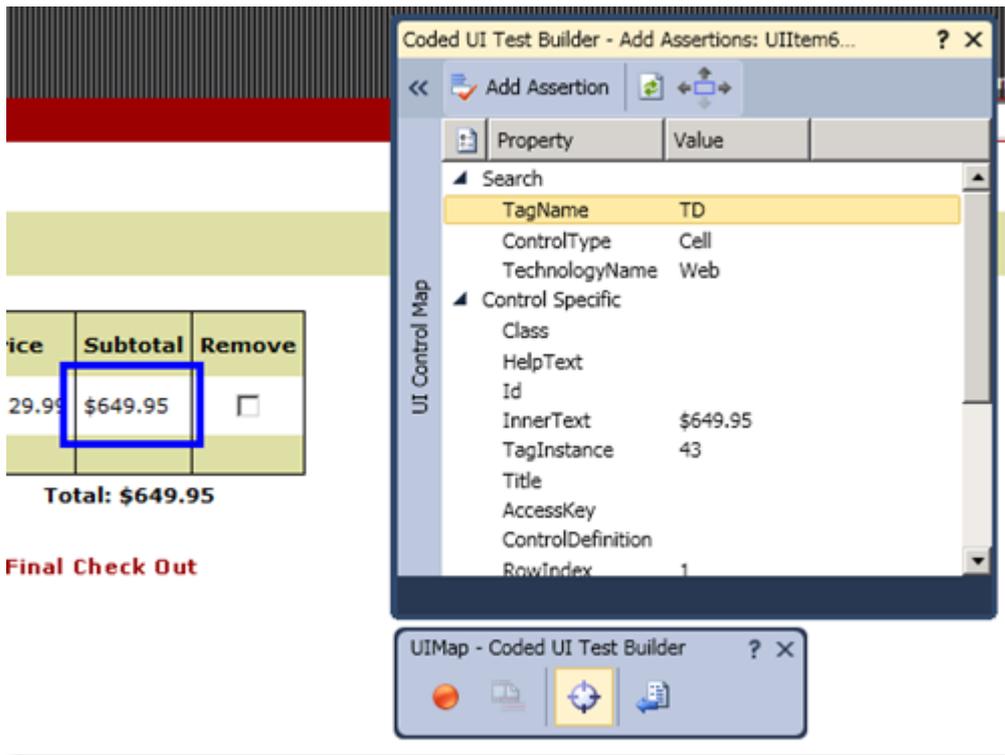


The Coded UI Test Recorder launched

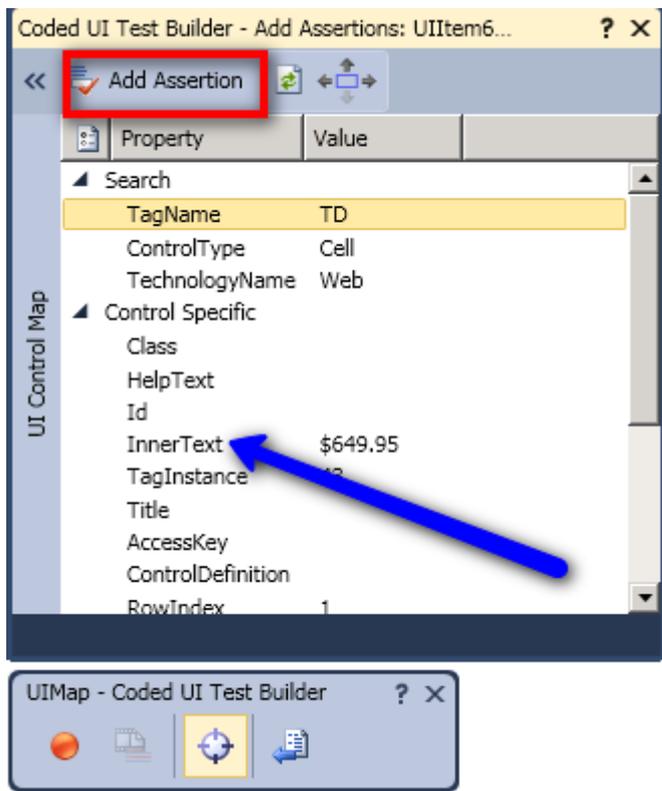


Step 3: Drag the cursor to the object you like to validate, in this case the “Subtotal”

The ID of the object is shown



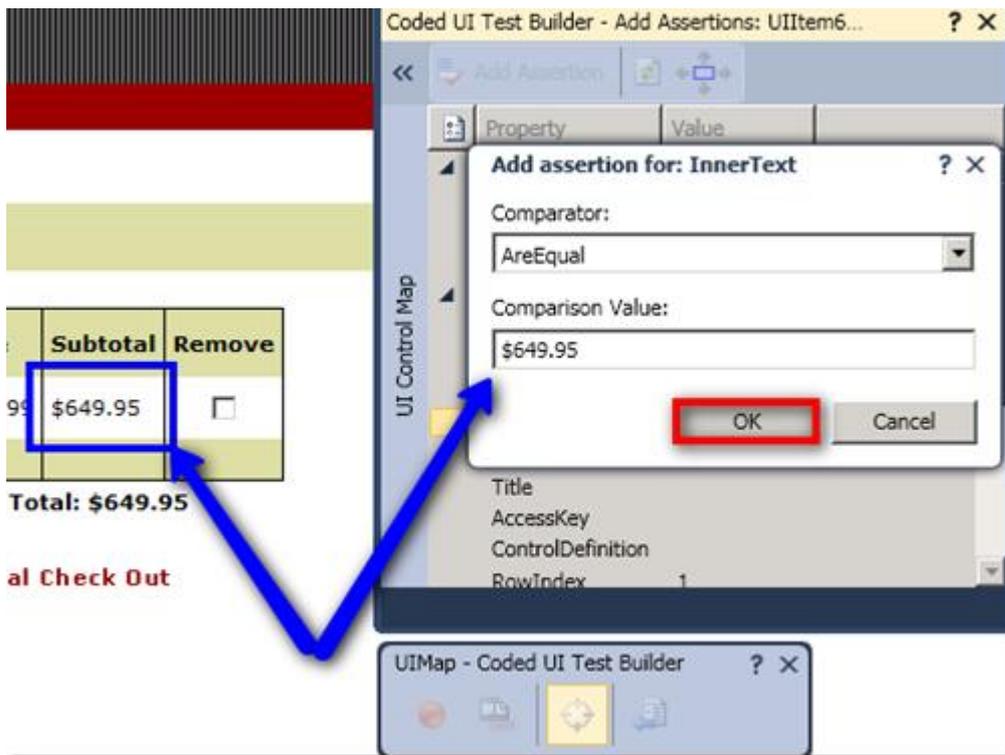
Step 4: Choose the property to validate and click: “Add Assertion”



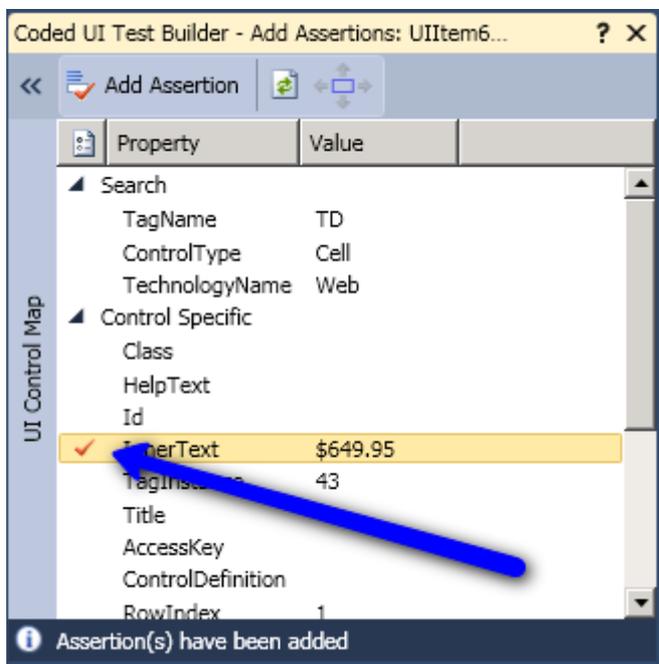
A new window pops up with the assertions available.

Step 5: Choose the Comparator and the Comparison Value

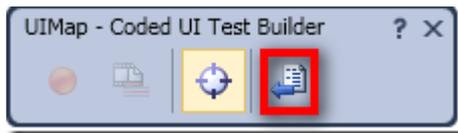
Step 6: Click OK



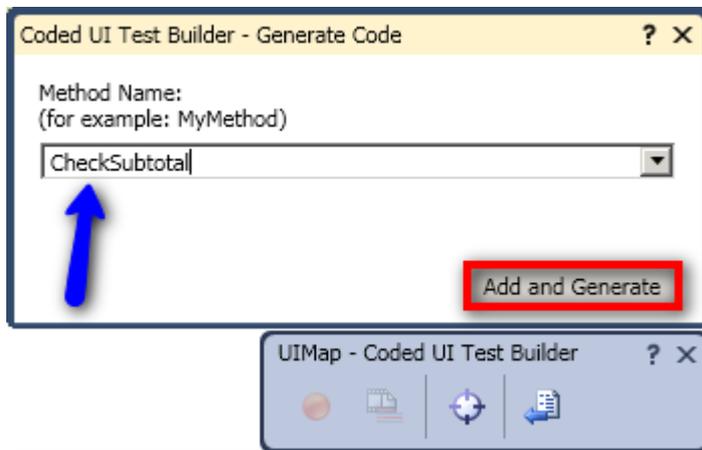
You will see a mark next to the property indicate that a validation rule was added



Step 7: Click the “Generate Code” button



Step 8: Type the name of the method and click “Add and Generate” button



You can see that the method was added to the “TestMethod” and will run with the test.

```
[TestMethod]
public void CodedUITestMethod1()
{
    // To generate code for this test, select
    // For more information on generated code
    this.UIMap.LaunchIBuySpy();
    this.UIMap.ClickOnMunitions();
    this.UIMap.AddMightyMightyPenToCart();
    this.UIMap.ChangeQuantity();
    this.UIMap.UpdateYourShoppingCart();
    this.UIMap.CheckSubtotal();
}
```

You can see that it is easy to add validations to the test, use validations as much as you can.

5. Summary

In this guide we have seen how to create Coded UI Tests.

You can see that it is very simple to create at least a basic Automation for your products.

I hope you liked this guide and learned how to have got more out of your TFS.

This guide was written based on a series of posts that I have published in the [My Blog](#) with small modifications.

Have Fun!!!