|  |
| --- |
|  |
| **Visual Studio** |
| **Coded UI for Microsoft Word 2010**  **Hands-on Lab (HOL)** |
|  |
| Thursday, June 14, 2012 |
| **Visual Studio ALM Rangers**  Andrew Whitechapel, Bijan Javidi, Ben Amodio, Bob Hardister, Mathew Aniyan, Mike Fourie, Willy-Peter Schaub |
| **Microsoft Corporation** |

|  |
| --- |
| **Visual Studio ALM Rangers**  This content was created by the Visual Studio ALM Rangers, a special group with members from the Visual Studio Product Team, Microsoft Services, Microsoft Most Valued Professionals (MVPs) and Visual Studio Community Leads. |

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft grants you a license to this document under the terms of the Creative Commons

Attribution 3.0 License**.** All other rights are reserved.

2012 Microsoft Corporation.

Microsoft, Active Directory, Excel, Internet Explorer, SQL Server, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

[Introduction 4](#_Toc327444125)

[Overview 4](#_Toc327444126)

[Visual Studio ALM Rangers 4](#_Toc327444127)

[Prerequisites 5](#_Toc327444128)

[Configuring a Virtual Machine 5](#_Toc327444129)

[Suggested Focus and Time Map 6](#_Toc327444130)

[Exercise 1: Install the Add-in and confirm installation 7](#_Toc327444131)

[Exercise 2: Set up the Test Project 10](#_Toc327444132)

[Exercise 3: Run the Coded UI Test for Microsoft Word 2010 11](#_Toc327444133)

[Exercise 4: Create a basic Coded UI Test for Microsoft Word 2010 13](#_Toc327444134)

[Exercise 5: Run the new basic Coded UI Test for Microsoft Word 2010 19](#_Toc327444135)

[Exercise 6: User Challenge 21](#_Toc327444136)

[Appendix 22](#_Toc327444137)

[Usage 22](#_Toc327444138)

[General Practices 22](#_Toc327444139)

[Custom Test Code 22](#_Toc327444140)

[References 22](#_Toc327444141)

[Visual Studio ALM Rangers Site 22](#_Toc327444142)

[MSDN Site 22](#_Toc327444143)

# Introduction

## Overview

The Visual Studio Coded UI Microsoft Word Add-in extends the Coded UI feature support to Microsoft Word 2010 documents.

Without the Coded UI Microsoft Word 2010 Add-in, the Coded UI feature:

* Can reference Word menu and ribbon controls
* Can only reference the screen location of mouse and keyboard actions within a document
* Does not support assertions within a document

The add-in provides the following additional functionality to Coded UI:

* Document location referencing of mouse and keyboard actions
* Assertions on contiguous document selection

## Visual Studio ALM Rangers

Visual Studio ALM Rangers is a special group with members from the Visual Studio Product group, Microsoft Services, Microsoft Most Valued Professionals (MVP) and Visual Studio Community Leads. Their mission is to provide out of band solutions to missing features and guidance. A growing [Rangers Index](http://blogs.msdn.com/b/willy-peter_schaub/archive/2010/06/18/introducing-the-visual-studio-alm-rangers-an-index-to-all-rangers-covered-on-this-blog.aspx) is available online[[1]](#footnote-2).

This guide targets the Microsoft “200-300 level” users of Visual Studio 2012. The target group is considered as intermediate to advanced users of Visual Studio and has in-depth understanding of the product features in a real-world environment. Parts of this guide may be useful to the Visual Studio novices and experts but users at these skill levels are not the focus of this content.

# Prerequisites

Although not required, we recommend that you perform this hands-on lab on a Virtual Machine environment.

### Configuring a Virtual Machine

There are several virtualization technologies and resources available to produce a virtual machine. We encourage you to look at the Rangers VM Factory if you find yourself producing virtual machines (or even physical) frequently. The project is hosted on CodePlex at <http://vsarvmfactory.codeplex.com>, and you can find an informative MSDN Magazine article ‘Visual Studio ALM Rangers: Rise of the VM Factory’ here: <http://msdn.microsoft.com/en-us/magazine/hh580740.aspx>. You may even prefer to download a recent Visual Studio virtual machine from Brian Keller[[2]](#footnote-3) and add any of the below listed pre-requisites which the virtual machine does not have.

A detailed walkthrough of creating a virtual machine is outside the scope of this hands-on lab. You will need the following to complete this hands-on lab.

**Operating System Platform**

* x86 or x64

**Operating System**

* Windows 8 Release Preview, Windows 7 SP1 or Windows Server 2008 R2

**Office Version**

* Microsoft Office 2010
* SP1 is recommended
* Only the 32 bit version is supported in this hands-on lab

**Development Tools**

* Visual Studio 2012 Ultimate or Premium
* WiX 3.6 RC0 - <http://wixtoolset.org/>
* Visual Studio Tools for Office 2010
  + Download from <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=20479>
  + Selected the x64 or x86 installer, based on the hardware configuration of your local machine

**Source Code and Installers**

* The Visual Studio Coded UI Word 2010 Add-in code can be downloaded from <http://go.microsoft.com/fwlink/?LinkID=237638>. To perform this lab you will need the following folders from the download:
  + CuiWord2010AddinSource
  + CuiWordAddinTestProject

## Suggested Focus and Time Map

If you intend to follow the hands-on lab (HOL) step-by-step, use these times as a guideline. If, however, you intend to investigate each step in detail double these times, at a bare minimum.

|  |  |  |
| --- | --- | --- |
| Topic | Duration in minutes | Page |
| Exercise 1 – Install the Add-in and confirm installation | 5 | 6 |
| Exercise 2 – Set up the Test Project | 10 | 10 |
| Exercise 3 – Run the Coded UI Test for Microsoft Word 2010 | 5 | 11 |
| Exercise 4 – Create a basic Coded UI Test for Microsoft Word 2010 | 15 | 12 |
| Exercise 5 – Run the new basic Coded UI Test for Microsoft Word 2010 | 5 | 18 |
| Exercise 6 – User Challenge | 15 | 12 |
| TOTAL | **55 min** |  |

*We wish you a pleasant and interesting journey!*

# Exercise 1: Install the Add-in and confirm installation

HinweisOBJECTIVE

In this exercise we install the add-in and confirm that it is installed correctly.

WARNING

To install the add-in, you must be logged on as an Administrator of the local machine.

1. Determine the correct MSI installer file according to the operating system platform of your local machine (x64 or x86).

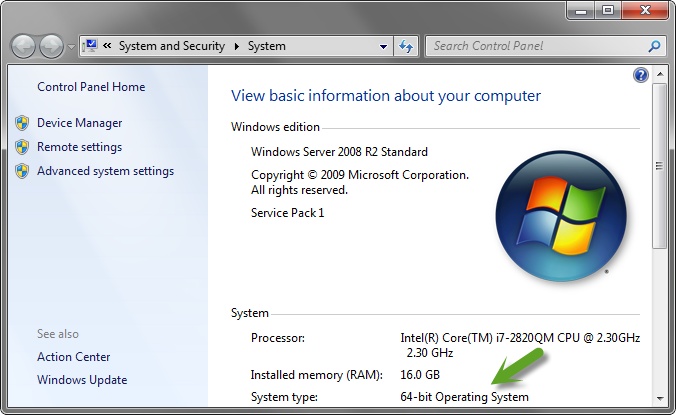


Figure – Determine your Operating System platform

1. Build the **CuiWordAddin.sln** in **Release** configuration to generate the MSI Installers.
2. Go to the output folder for the applicable flavor installer you need and right-click the MSI installer file and then click **Install.** If you are running a 64-bit operating system, then you should run Setup.CuiWordAddin.**x64**.msi. If you are running a 32-bit operating system, run Setup.CuiWordAddin.**x86**.msi

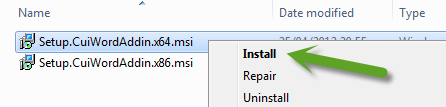


Figure – Install the MSI

1. Complete the installation wizard.
2. Click the **Start** button on the Windows taskbar.
3. Type **word.**
4. Press **Enter** to launch Microsoft Word 2010.

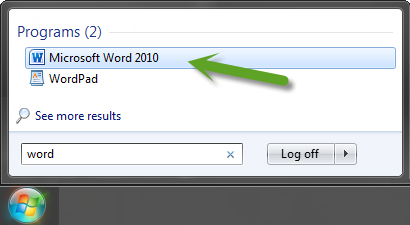


Figure – Start Microsoft Word

1. If a confirmation window appears, click **Install** to complete the installation of the add-in.

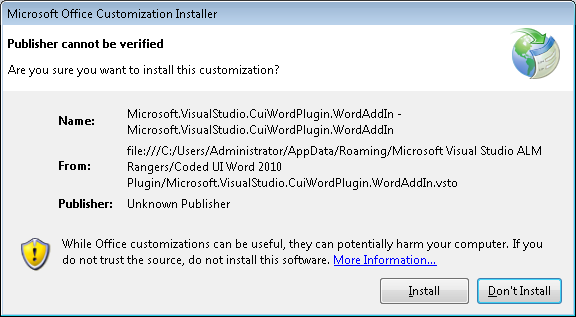
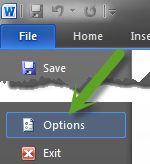


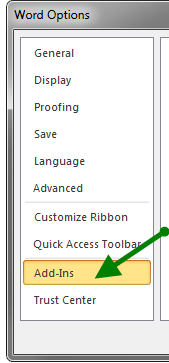
Figure – Install the add-in

1. Click the **File** tab on the top of the Word ribbon.
2. Select **Options.**



**Figure 5 – File: Options**

1. Select **Add-Ins**



**Figure 6 – Word Options: Add-Ins**

1. Ensure that COM Add-ins is selected in the Manage field at the bottom of the window.
2. Click the Go button to the right of the Manage field.

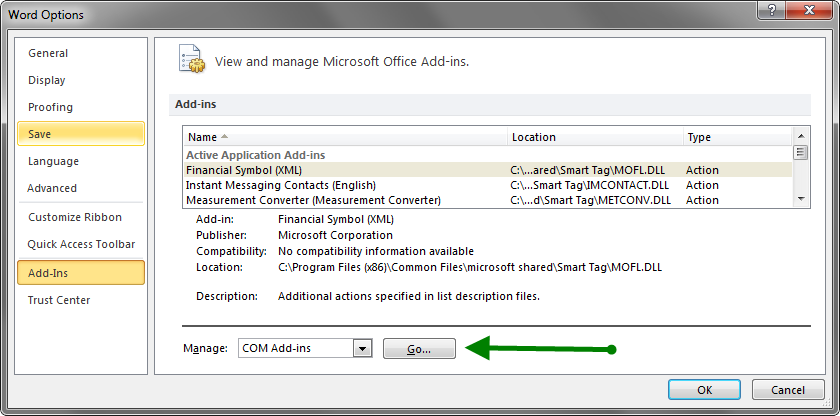


Figure – Manage COM Add-ins

1. Ensure the Add-in **Microsoft Visual Studio Coded UI Word 2010 Add-in** is listed and the check box is selected.

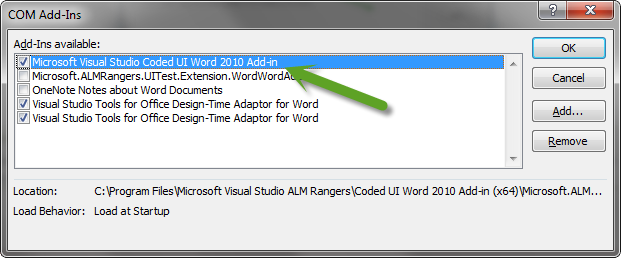


Figure – Manage COM Add-ins

HinweisREVIEW

We have installed and verified that the Coded UI for Microsoft Word 2010 Add-in was installed correctly. The next step is to set up the test project contained in the sample Coded UI test.

# Exercise 2: Set up the Test Project

HinweisOBJECTIVE

In this exercise we open the test project in Visual Studio.

1. Copy the **CuiWordAddinTestProject** folder from the download location on your local machine to a location suitable for your Visual Studio solution and project files.
2. Start **Visual Studio**
3. Open the **CuiWordAddinTestProject** in Visual Studio. On the **File** menu, click **Open**, **Project/Solution**, [your Visual Studio project folder], **CuiWordAddinTestProject.sln**.
4. Press **F6** to build the project.
5. Launch Microsoft Word (if not already running).
6. Size Word and Visual Studio windows to fit side by side on your screen.

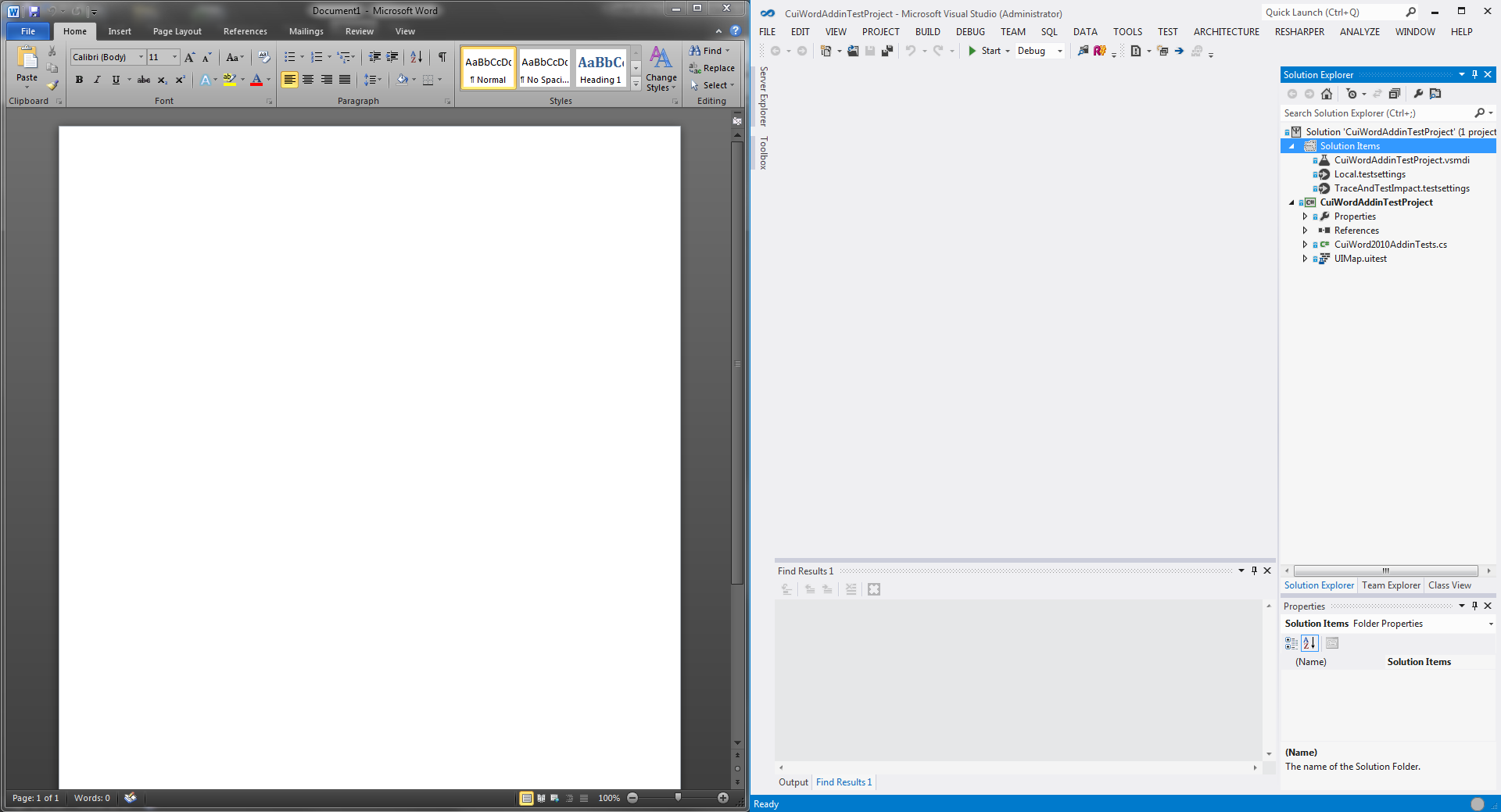


Figure – Prepare for testing

1. Close Microsoft Word. The sample automated Coded UI test expects this.
2. Open the Visual Studio **Test Explorer**.

(On the **Unit Test** menu, point to **Windows**, and then click **Test Explorer**)

HinweisREVIEW

We have set up the Coded UI test project. The next step is to run the sample Coded UI for Microsoft Word tests.

# Exercise 3: Run the Coded UI Test for Microsoft Word 2010

HinweisOBJECTIVE

In this exercise we execute the sample Coded UI test for Microsoft Word 2010.

WARNING

No instances of Microsoft Word should be running. If you are currently using this HOL in Word, you need to close it.

1. Build the solution to get the unit test(s) listed in the **Test Explorer**
2. In the Unit Test Explorer, right-click the **RecordAndSelectText** test and click **Run Selected Tests**.

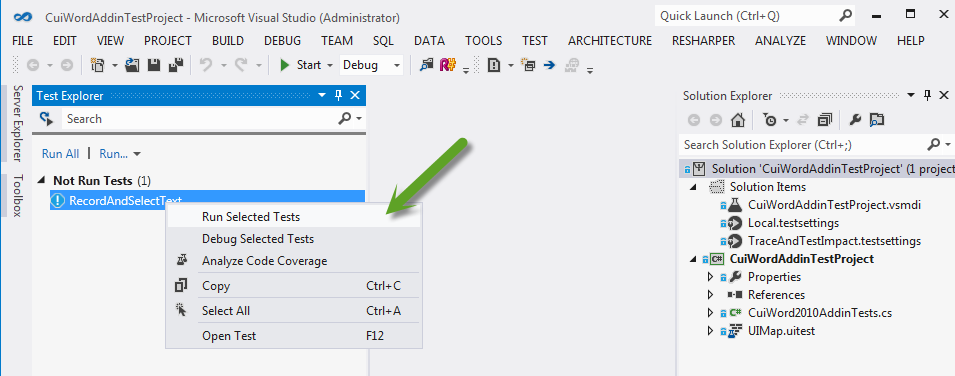


Figure – Run the test

1. The test will open Microsoft Word, enter text, select text and complete.

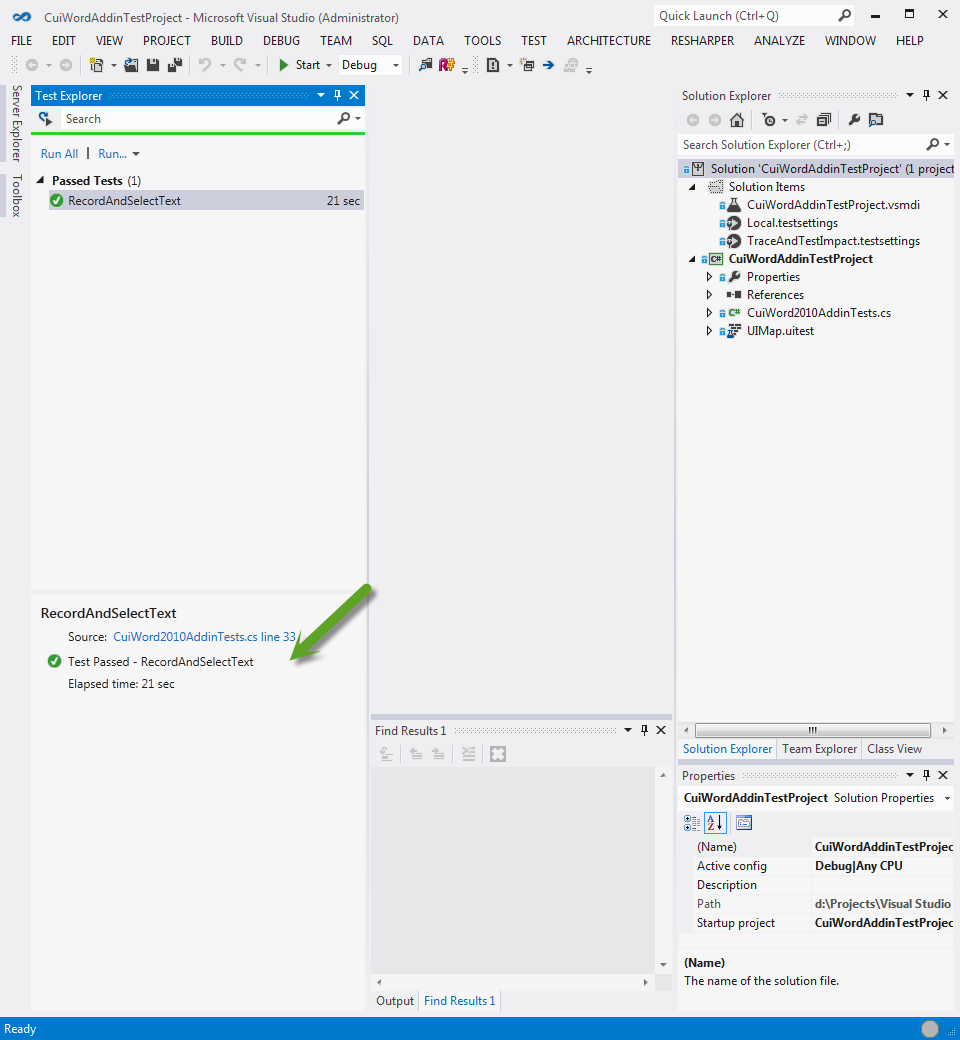


Figure – Test Execution

NOTE

A change in Visual Studio 2012 is that the ApplicationUnderTest, in this case Microsoft Word, is closed automatically after the test is run. At the time of compiling this hands-on lab the MSDN documentation incorrectly states that the application is left open as was the case for Visual Studio 2010.

HinweisREVIEW

We have tested the Coded UI Microsoft Word 2010 Add-in using the example Coded UI test. The next step is to create a Coded UI test for Microsoft Word 2010.

# Exercise 4: Create a basic Coded UI Test for Microsoft Word 2010

HinweisOBJECTIVE

In this exercise we create and execute the example Coded UI test for Microsoft Word 2010.

1. On the **View** menu, select **Solution Explorer** to display the CuiWordAddinTestProject solution.

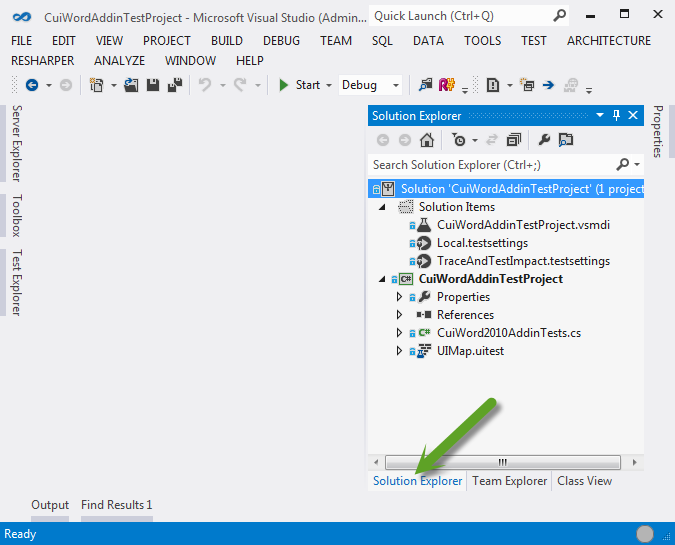


Figure – Solution Explorer

1. Right-click the project under the solution.
2. Click **Add**, **New Item…**.

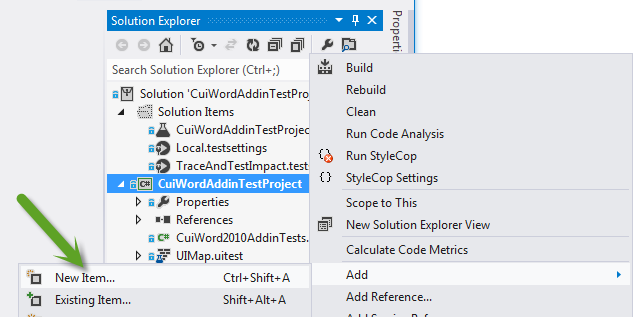


Figure – Add a New Test

1. Select **Coded UI Test…** and call it RecordAndSelectText2.cs
2. Type **RecordAndSelectText2** in the **Test Name** text box and click **OK**.

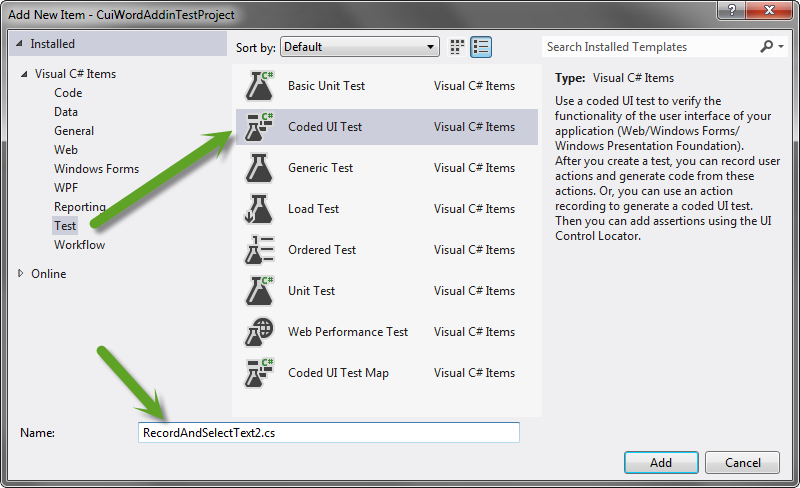


Figure – Add new Coded UI test

NOTE

The **Generate Code for Coded UI Test** dialog box should now appear (it may take a few seconds)

1. Verify that the **Record actions, edit UI map or add assertions** radio button is selected on the **Generate Code for Coded UI Test** dialog box and click **OK**.

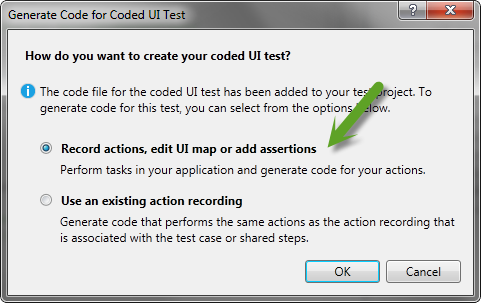


Figure 15 – Generate Code for Coded UI Test dialog

NOTE

The **You have made changes to your tests** dialog box might appear. If it does, click **OK**.

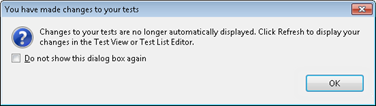
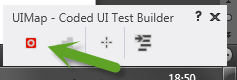


Figure 16 – You have made changes to your tests dialog

NOTE

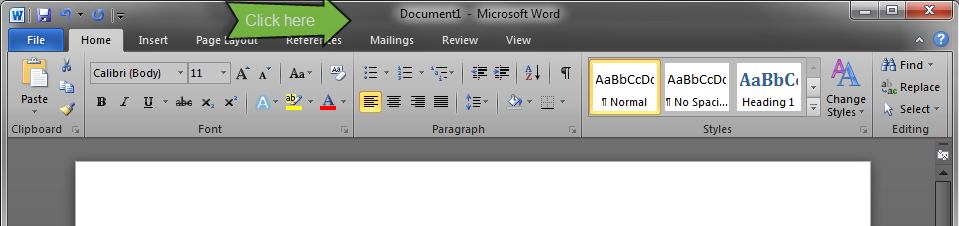
The **UIMap – Coded UI Test Builder** gadget should now display in the lower right portion of your screen.

1. Click the red **Start Recording” button on the “UIMap – Coded UI Test Builder** gadget.



**Figure 17 – Coded UI Test Builder gadget**

1. Click the top title bar of the open Microsoft Word document.



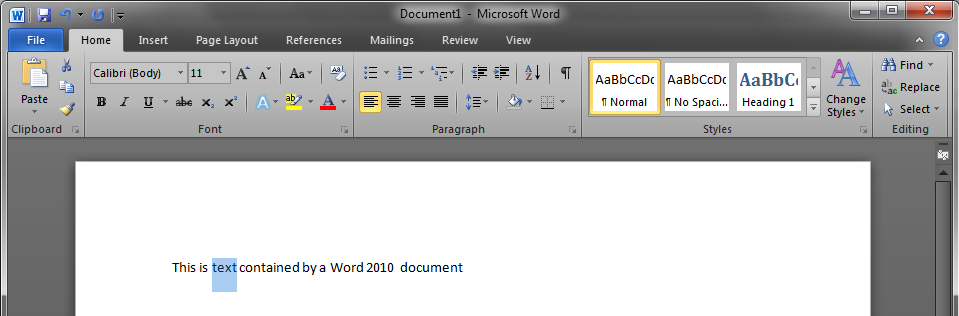
**Figure 18 – Select the document title bar**

1. In the document, type “This is text contained by a Word 2010 document.”
2. Press and hold the CTRL key and press the Left Arrow key until the cursor is just to the left of the “t” in the word “text.”

NOTE

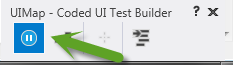
The next several steps use generic Coded UI features to create the test. The add-in leverages existing Coded UI feature functionality. Review the “Overview” section above to learn what distinct features are provided by the add-in vs. generic Coded UI feature functionality.

1. Press and hold the SHIFT and the CTRL keys and press the Right Arrow key one time to select the word “text.”



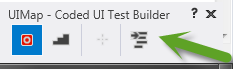
**Figure 19 – Select text**

1. Click the pause button on the **UIMap – Coded UI Test Builder** gadget to pause the recording



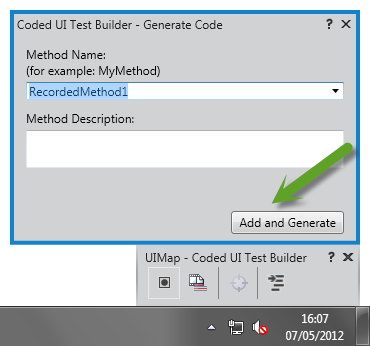
**Figure 20 – Pause the recording**

1. Click the **Generate Code** button on the **Coded UI Test Builder** gadget.



**Figure 21 – Generate Code button**

1. Click the **Add and Generate** button on the **Coded UI Test Builder - Generate Code** dialog box.



**Figure 22 – Add and Generate the code**

1. Using the mouse, select the **crosshair** icon, and holding down the mouse key, drag the Crosshair button on the **Coded UI Test Builder** gadget over the selected word “text” in the Microsoft Word document

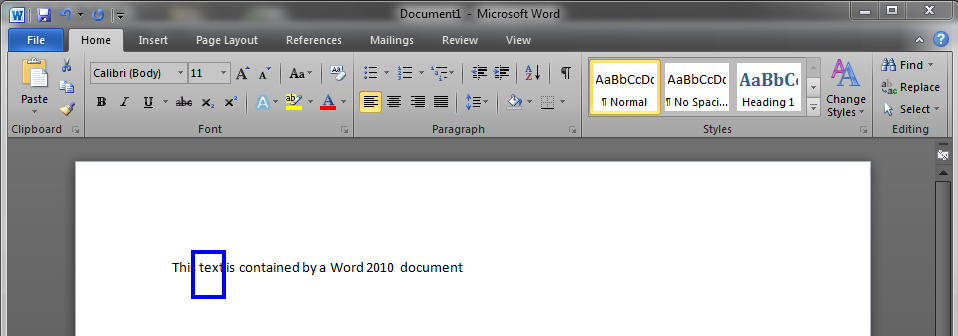


**Figure 23 – Coded UI Test Builder gadget**

1. A **blue box** should appear around the word “text” inside the document.

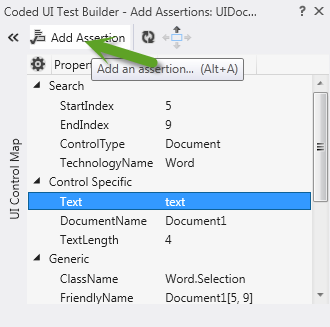
***The ability for the crosshair to identify objects within a document is provided by the add-in.***

Release the mouse



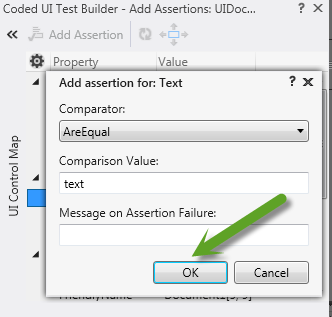
**Figure 24 – Blue box provided by Add-in**

1. Click the **Text** property within the **Coded UI Test Builder – Add Assertions** dialog box.
2. Click the **Add Assertion** button on the **Coded UI Test Builder – Add Assertions** dialog box.



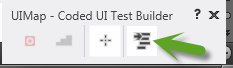
**Figure 25 – Add Assertion**

1. Click **OK** on the **Add assertion for: Text** dialog box.



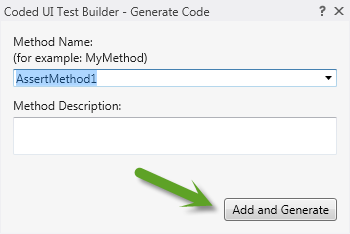
**Figure 26 –Assertion details**

1. Click the **Generate Code** button on the **Coded UI Test Builder** gadget.



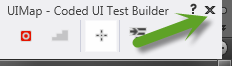
**Figure 27 – Generate Code button**

1. Click the **Add and Generate** button on the **Coded UI Test Builder = Generate Code** dialog box.



**Figure 28 – Generate assertion code**

1. Click the “x” on the upper-right corner of the **Coded UI Test Builder** gadget to close it and save the test.



**Figure 29 – Close the gadget**

1. Close Microsoft Word and do not save the document.

HinweisREVIEW

We have just created a new Coded UI Test using the Coded UI for Microsoft Word 2010 Add-in. The next step is to run the new test.

# Exercise 5: Run the new basic Coded UI Test for Microsoft Word 2010

HinweisOBJECTIVE

In this exercise we run the new Coded UI test for Word 2010.

1. Open the RecordAndSelectText2.cs file.
2. Rename the Test Method from “CodedUITestMethod1()” to “RecordAndSelectTextTwo()”

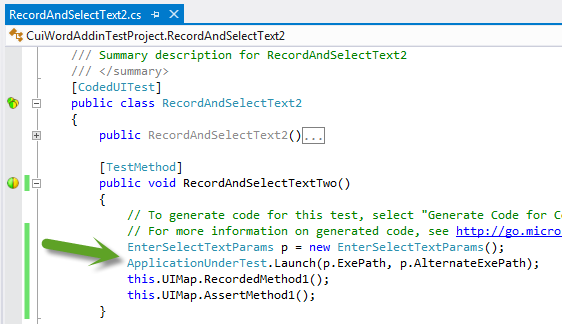


**Figure 30 – Rename the test**

1. We need to ensure that Word is started. Add the following code to your Test Method

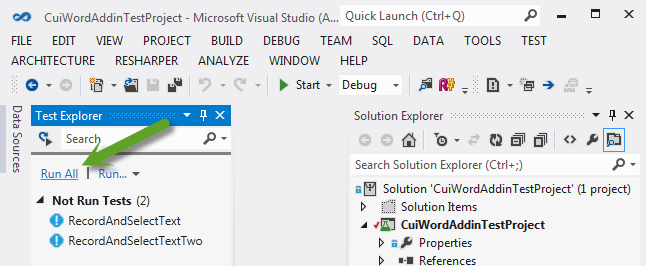
EnterSelectTextParams p = new EnterSelectTextParams();

ApplicationUnderTest.Launch(p.ExePath, p.AlternateExePath);



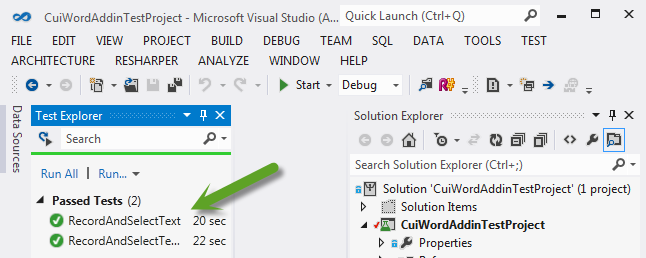
**Figure 31 – Add code to start Microsoft Word**

1. Press **F6** to build the project.
2. Click **Run All** in Text Explorer.



**Figure 32 – Run both tests**

1. Confirm that both the example test and the new test executed successfully.



**Figure 33 – Tests run and passed**

HinweisREVIEW

We have created and executed a new Coded UI Test using the Coded UI Test for Microsoft Word 2010 Add-in. Please feel free to review the generated code, modify these tests, and create new ones. See the Usage section below for some general practices and guidelines for creating Coded UI tests using the Coded UI Test for Microsoft Word 2010 Add-in. See the References section below for links to comprehensive guides on creating and using the Coded UI Test feature.

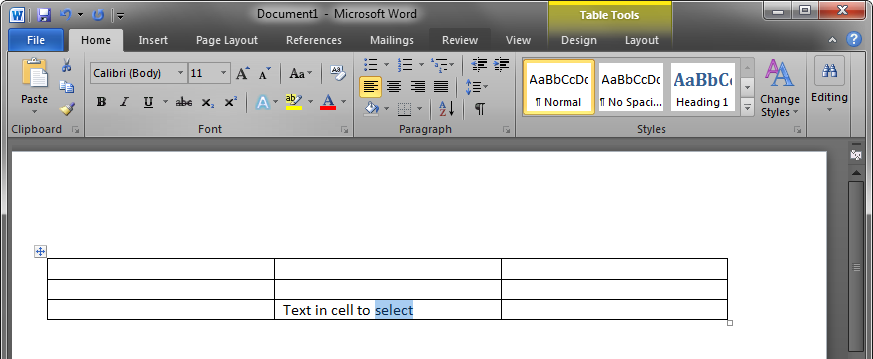
# Exercise 6: User Challenge

HinweisOBJECTIVE

In this exercise we challenge you the user to Create a Coded UI Test for Microsoft Word 2010 with tables

Create a Coded UI test to

* create a word document
* create a table
* enter text in the table
* validate the word selected, as shown in the sample below



If you encounter any difficulties with this challenge, join us in discussion online at our CodePlex home[[3]](#footnote-4). We welcome and appreciate your feedback.

# Appendix

## Usage

This section provides a brief summary on developing a Coded UI Test using the Coded UI Word 2010 Add-in. Please see the Reference section below for links to comprehensive guidance on Coded UI Test.

### General Practices

* Use the keyboard for all actions within a Word document surface when possible (instead of the mouse).
* Always click the tab or the window header before clicking or typing within the tab or the window.
* Break up and group steps into small methods for reuse and stability (isolate fragile code).
* If the test code fails to execute correctly, change the recording sequence and/or approach you are using
* Turn off Word proofing options.
* If prompted, do NOT normalize line endings.

### Custom Test Code

* To avoid having your customization overwritten next time you generate code using the Coded UI Test Builder, copy the recorded method from the UIMap.Design.cs file into the UIMap.cs file and rename to {name}Custom.
* Methods not impacted by application changes (that is, start Word).
* Use CUI WaitForControlExist(), WaitForControlNotExist(), etc., methods when needed to ensure that the next control you want to click is available. If Coded UI fails because the control is not available (you’ll hear a beep), failure of a subsequent step is likely.

## References

### Visual Studio ALM Rangers Site

<http://msdn.microsoft.com/teamsystem/ee358786.aspx>

<http://www.tinyurl.com/almrangers>

### MSDN Site

http://msdn.microsoft.com/en-us/library/dd286726.aspx

<http://msdn.microsoft.com/default.aspx>

1. Rangers Index - <http://blogs.msdn.com/b/willy-peter_schaub/archive/2010/06/18/introducing-the-visual-studio-alm-rangers-an-index-to-all-rangers-covered-on-this-blog.aspx> [↑](#footnote-ref-2)
2. <http://blogs.msdn.com/b/briankel> [↑](#footnote-ref-3)
3. <http://vsarcodeduiword.codeplex.com/discussions> [↑](#footnote-ref-4)