#### C:\Users\tstar\VS2012.ALMRangers.Logo.NoTrademark.Transparent.Purple.1416x475.png

Epic: As Doris, the developer, I would like to understand how to build Coded UI Tests that can be shared across UI components for unit testing.

# Overview

The Coded UI test code generating capabilities in Visual Studio do a fine job of generating code from action recordings or using the Coded UI test builder, but if you do not take control of your test projects using multiple UI maps you will have some challenges maintaining your test projects.

Multiple UI Maps allow you to better organize your UI testing assets to make testing a large application with a team approach much easier. It will require some code on your part, but much of the glue to support multiple UI maps and more robust assertions is straightforward and does not require an expert command of your .Net language of choice.

# Key Features

* **Create Multiple UI Maps** to better support a more modular approach to testing. By creating a UI map per complex control or page the testing team can be working on different UI maps at the same time without worrying about conflicts with other team members. Multiple UI maps will make your test project easier to fix in the case of a breaking change to the UI because you do not have all the UI code in one giant monolithic UI Map. *Additionally, by using the Coded UI Test builder, you can easily add new UI controls to an existing map.*
* **Use the UI Map Editor** to move the code you want to modify out of the designer and into the code file so changes are not lost the next time the UI map is generated.
* **Override or Customize Assertions** dynamically change the recorded value to a test specific value. You can also write completely custom assertions that call numerous other assertions within a map so the Coded UI test cases can validate an entire page, complex control, or some portion thereof with a single call.

**Creating Multiple UI Maps**

|  |  |  |
| --- | --- | --- |
| **1** | In the test project, Add – New Item – Coded UI Test Map. | 🞏 |
| **2** | Give your map a logical name, perhaps the page name or complex control name. | 🞏 |
| **3** | Use the coded UI Test Builder to generate the UI Map. | 🞏 |
| **4** | Create Or Modify a utility class to expose the UI map to your Coded UI tests. | 🞏 |

**Use the UI Map Editor To**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | Deleted generated code or navigation steps. | | 🞏 |
| **2** | Rename methods and assertions. | | 🞏 |
| **3** | Move code to the editable code location. | | 🞏 |
| **4** | Insert delays. | | 🞏 |
| **5** | Split methods. | | 🞏 |
| **6** | Locate UI controls. | | 🞏 |
|  | |

**Override and Customize Assertion**

|  |  |  |
| --- | --- | --- |
| **1** | Use a single line of code to alter the expected value used in an assertion. | 🞏 |
| **2** | Write a custom assertion method to encapsulate calling multiple assertions on a single control or page. | 🞏 |
| **3** | Use databinding techniques to make your assertions more robust. | 🞏 |

#### BUILDING CODED UI MAPS TO SHARE ACROSS TESTS.

**References** Visual Studio ALM Rangers, Copyright 2012 Microsoft Corporation

|  |  |  |  |
| --- | --- | --- | --- |
| <http://blogs.msdn.com/b/anutthara/archive/2010/02/08/scaling-up-your-cuit-ui-automation-for-real-world-projects.aspx> | <http://msdn.microsoft.com/en-us/library/ff398056.aspx> | <http://blogs.msdn.com/b/anutthara/archive/2010/02/10/walkthrough-using-multiple-coded-ui-maps-in-test-automation.aspx> |  |
|  | http://blogs.msdn.com/b/anutthara/archive/2010/02/10/walkthrough-using-multiple-coded-ui-maps-in-test-automation.aspx |  |  |
|  |  |  |  |