



# Test Automation FX

UI test automation for .NET developers

## Using Test Automation FX with a Test First approach

### Example

This example shows how we can develop a simple application with a test first approach using TAFX.

First we have to have a general idea of what the application will look like and what it will do. For this example we will try to develop a simple dialog that will allow the user to enter his/her name in the form first name and last name. We would also want to have an additional field that shows the user's full name as a result of the two previous fields.

First name:

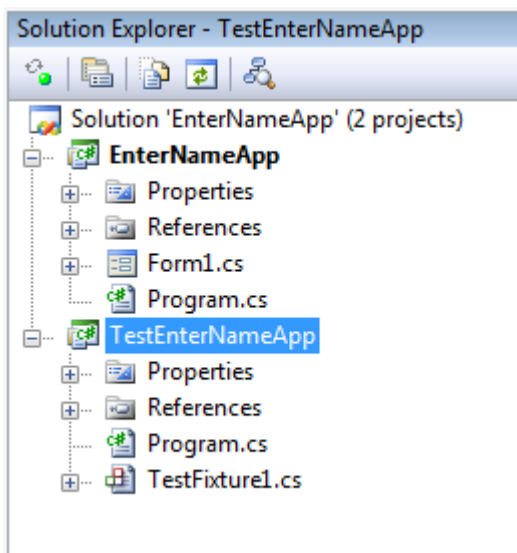
Last name:

Full name:

Picture 1 A sketch of the application we are going to build.

Let's create a solution in Visual Studio that contains two projects:

1. One project named EnterNameApp that is a simple windows forms application.
2. One Test Automation FX Test application.



Picture 2 The solution explorer after we have created a windows forms application and an accompanying test application.

As we are going to use a test first approach, we will start with the test application. We add the user interface elements that we expect to have in the application. Since we do not have

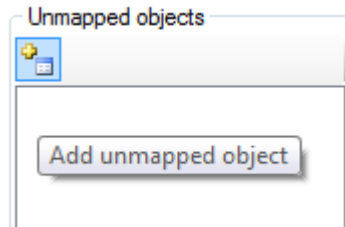




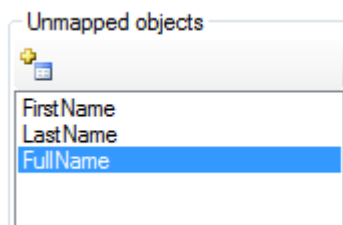
# Test Automation FX

## UI test automation for .NET developers

an application to map objects from at the moment, we add some unmapped objects to the TAFX UIMap.

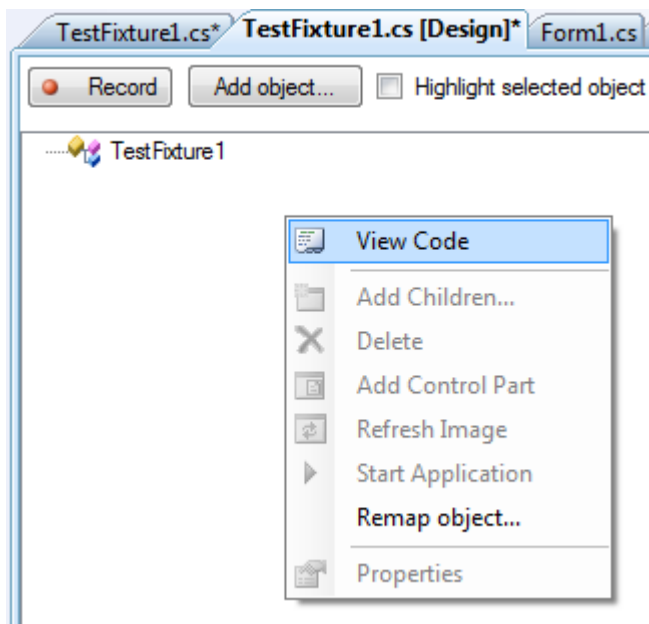


We add the three elements that we need for our initial test.

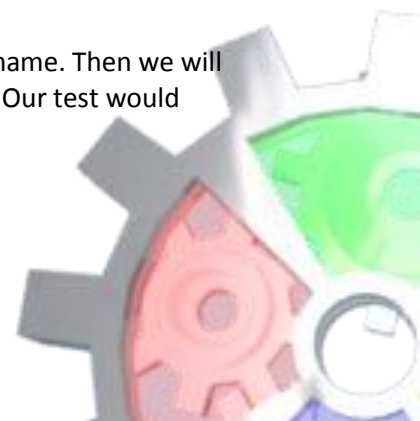


*Picture 3 The unmapped object container in the UIMap designer. We have added the three objects that we will be using in our test.*

Now we can start writing our test using these objects. Select **View Code** and add a test method containing the test logic.



We have to add some sample name in the text boxes for the first and last name. Then we will have to verify that the textbox for the full name contains the whole name. Our test would then look something like:



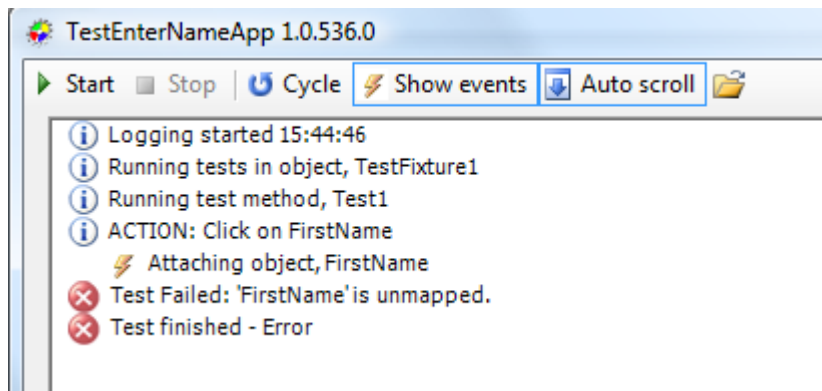


# Test Automation FX

## UI test automation for .NET developers

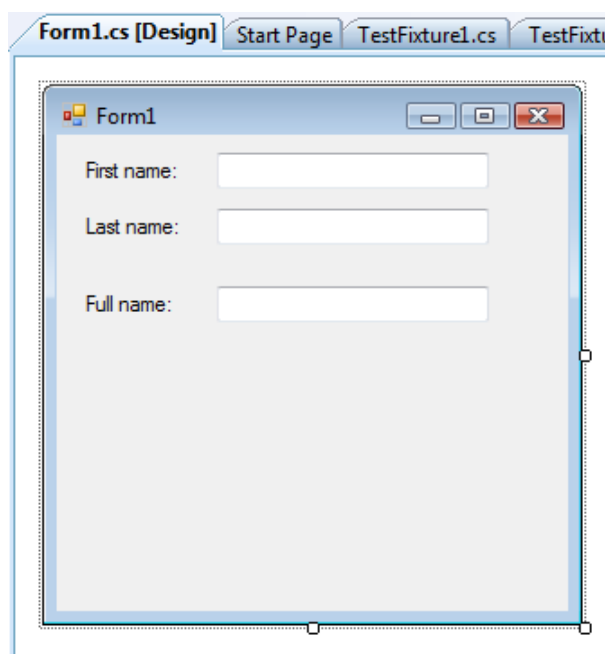
```
[UITest()]\n\npublic void Test1()\n{\n    FirstName.EnterText("Tom");\n    LastName.EnterText("Jones");\n}
```

Now we can run the test. But the test will fail since it contains unmapped objects. A test containing an unmapped object will fail when any of those objects are needed to perform some action.



Ok, so now it is time to start implementing a user interface for our application under test.

Let's bring up the Form1.cs class in our EnterNameApp project and add some controls.

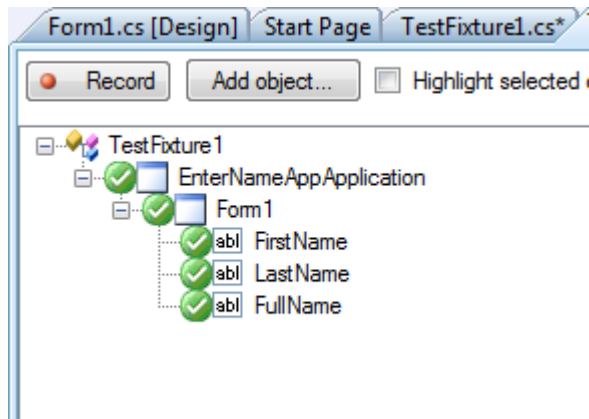




# Test Automation FX

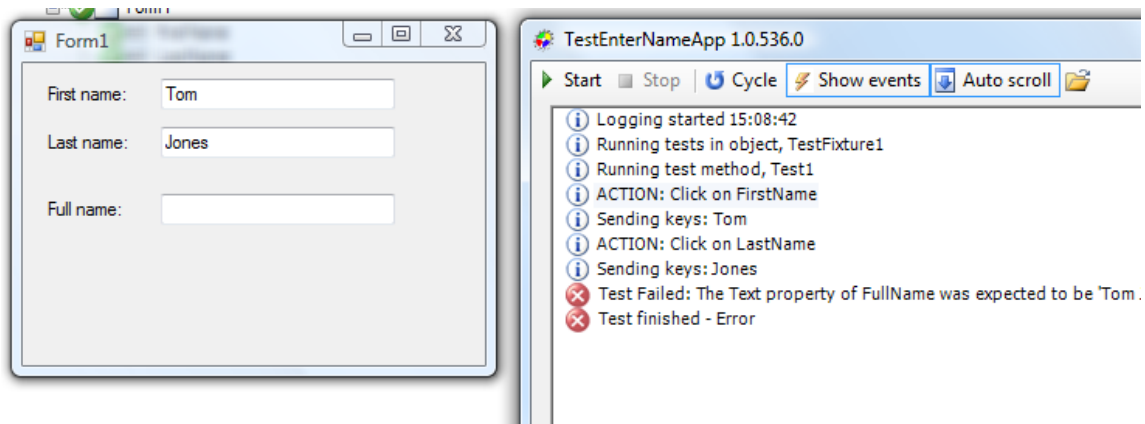
## UI test automation for .NET developers

We simply add these controls to the form and start the application. We will use the context menu command 'Map object' of the unmapped objects to point to the text boxes of the running application. As they are mapped, the unmapped objects are moved into the UIMap tree under the application.



Picture 4 The UIMap after the objects have been mapped to the actual controls.

Now, if we try to run the test we will get an error again, but this is because we still miss some of the functionality we defined in the test.

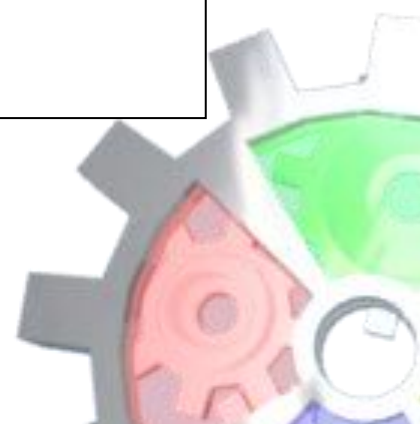


The full name text box does not contain its expected value and thus the test fails.

We implement this functionality by adding an event handler for the TextChanged event for the first name textbox. We connect the last name textbox to the same eventhandler.

```
private void OnNameChanged(object sender, EventArgs e)
{
```

Running the test again gives us a better result!





# Test Automation FX

UI test automation for .NET developers

The screenshot displays two windows. The left window, titled 'Form1', contains three text input fields: 'First name:' with 'Tom', 'Last name:' with 'Jones', and 'Full name:' with 'Tom Jones'. The right window, titled 'TestEnterNameApp 1.0.536.0', shows a test execution log with the following entries:

- Start
- Stop
- Cycle
- Show events
- Auto scroll
- Logging started 15:19:06
- Running tests in object, TestFixture1
- Running test method, Test1
- ACTION: Click on FirstName
- Sending keys: Tom
- ACTION: Click on LastName
- Sending keys: Jones
- The Text property of FullName was verified to be 'Tom Jones'
- Test finished - Success

