# BTSHOL04: Unit Test and Deployment

Overview

This lab will demonstrate the unit testing features of BizTalk Server 2016 and Visual Studio 2015. Specifically, you will see how to create unit test for schemas and maps. Previous labs has shown how schemas and maps can be tested using the Visual Studio user interface. This however can sometimes be a tedious task that can take time when you have lots of artifacts and want to test them due to some minor changes being made. In those scenarios BizTalk artifacts can be tested using the Visual Studio Test tools and their continuous compatibility when changes occur can be easily controlled.

The lab will also show you how to deploy a project, configure and start it once deployed.

Objectives

After completing this lab, you will be able to:

* Enable Unit Testing for BizTalk projects
* Create a Unit Test project in Visual Studio
* Add code to test BizTalk artifacts
* Deploy a BizTalk project from Visual Studio
* Use the BizTalk Server 2016 Administration Console to verify deployment

Scenario

As part of their internal initiative to produce better quality development services and setting up continuous integration for their BizTalk Server division Northwind Traders has asked you to implement unit tests for your BizTalk artifacts.

First, you will assign enable using testing for your BizTalk project. Then you will create a test project to enable you to write tests. You will need input and output files to compare with to determine whether the test has been successful or not. After that you will add code to test your artifacts against the files and verify the result. Once you have verified that the test succeeds you will configure your project to be deployed and deploy it using Visual Studio. This will deploy the application to BizTalk Server 2016.

Estimated time to complete this lab:   
Exercise 1-4: 30 minutes  
Exercise 5-6 10 minutes

User Name: **Administrator**

Password: **pass@word1**

Exercise 1

Enable Unit Testing on BizTalk projects

Unit tests are methods triggered by a Test tool to test components at a lower level then project or assembly level – early in the development process. Identifying issues with your artifacts early saves time and effort involved when discovering errors late. In this exercise, you will open an existing BizTalk Server project and alter its settings to enable unit tests.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Open the BizTalk Server 2016 Solution. | 1. In Windows Explorer, browse to **C:\Labs\Lab 4\Start\NWBusinessSolution**,and then double click **NWBusinessSolution.sln** to open the solution.   *The provided solution contains a few schemas and a map. You will add tests for a few of these.* |
| 1. Configure the BizTalk Server project.   Throughout this lab you will test and eventually deploy this project. | 1. In Solution Explorer, right click the **NWMessaging** project and select properties. 2. In the left pane of **Properties** window,select **Deployment**..   *These properties represent those that you can set related to deploying your BizTalk project.*   1. Under **Unit** Testing, select **Enable Unit Testing**. 2. Set it to **True**.   *This enables unit testing by altering the default build process for BizTalk artifacts to include test methods.*   1. Close the property pages saving the changes. 2. Build the project.   *This will build the project and prepare it to be tested.* |

Exercise 2  
Add a Test project

In this exercise, you will add a Test project to your solution. You will then add references to BizTalk Server assemblies. Finally you will write the code that performs the tests.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create the test project. | 1. On the **NWBusinessSolution** file right click add new project 2. Select **Test** from the Templates, and then select **Unit test project**, and name it **NWTest** click Ok.   *This will create the project and add it to the solution explorer* |
| 1. Add items, add references. | 1. Delete the auto generated **UnitTest1.cs** 2. On the **NWTest** project right click – **add new item.** 3. Select **Test** from the Templates, select Unit Test, name it **CustomerOrderTest.cs** click Add. 4. Under NWTest select References – right click and **add** following references:   “**NWMessaging.dll**” (find it under “solution” in reference manager) click OK.   **Microsoft.BizTalk.TestTools.dll** (in reference manager type “Biztalk” in the search assembly and select the dll. Click OK.  **Microsoft.XLANGs.BaseTypes.dll** (reference manager type “Xlang” in the search assembly). and select the dll. Click OK. |

Exercise 3  
Test a schema

In this exercise, you will use add code to the **CustomerOrderTest** class to test the schema.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Define the test.  You add code to point out an input file, verify that that file exists and use the schema to validate that the file conforms to the schema. | 1. From the **NWTest** project, open the **CustomerOrderTest.cs** file. 2. Add the following code at the top of the file:   using System.IO;  using NWMessaging;   1. Rename the auto generated method:   public void TestMethod1()  to:  public void CustomerOrderConstructorTest()   1. In the new method – replace below comment   //  // TODO: Add test logic here  // with following code:  CustomerOrder target = new CustomerOrder();  string schemaInstance = @"C:\Labs\Lab 4\Start\CustomerOrder.xml";  Assert.IsTrue(File.Exists(schemaInstance));  Assert.IsTrue(  target.ValidateInstance(schemaInstance,Microsoft.BizTalk.TestTools.Schema.OutputInstanceType.XML));   1. Compile the **NWTest** project and ensure that it builds successfully. |
| 1. Run the test | 1. From the **Test** menu, select **Windows** – **Test Explorer.** 2. In the **Test Explorer View**, select **CustomerOrderConstructorTest**. 3. Right click it and select **Run Selection**. 4. Verify that the test got a result of **Passed**. |

Exercise 4  
Test a map

In this exercise, you will add code to you the **NWCustomer\_to\_FKSupplierPOTest** to test the map.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Define the test.  You add code to feed an input file into a map, select the location for an output file that will get validated for correctness, and make sure that the output file could be created. | 1. Select the **Solution Explorer**, find the **NWTest** project and open the **NWCustomer\_to\_FKSupplierPOTest** class. 2. Find the **NWCustomerOrder\_To\_FKSupplierPOConstructorTest()** method. 3. Initialize the sourceFile variable to C:\Labs\Lab 4\Start\CustomerOrder.xml. Like so string sourceFile = @"C:\Labs\Lab 4\Start\CustomerOrder.xml"; 4. Add a new variable for the output file string destFile = @"C:\Labs\Lab 4\Start\SupplierPO.xml"; 5. Add code to test the map target.ValidateOutput = true; target.TestMap(sourceFile, Microsoft.BizTalk.TestTools.Schema.InputInstanceType.Xml, destFile, Microsoft.BizTalk.TestTools.Schema.OutputInstanceType.XML); 6. Add code to verify the output of the test Assert.IsTrue(File.Exists(destFile)); |
| 1. Run the test | 1. From the **Test** menu, select **Windows** – **Test View.** 2. In the **Test View**, select **NWCustomerOrder\_To\_FKSupplierPOConstructorTest(**. 3. Right click it and select **Run Selection**. 4. Verify that the test got a result of **Passed**. |

Optional  
Help, Samples and Testing a pipeline

This optional exercise will cover two interesting aspects. The first is the BizTalk Server downloadable help file, which allows you to search the over 8000 pages of BizTalk help in a focused manner eliminating the noise of the internet. This gets you search hits limited to BizTalk Servers official help files, but at the same time you loose the results from sources like forums and blogs etc. The second is the knowledge about the SDK directory of BizTalk Servers installation directory, and the wealth of samples that can be found there.

If you have not already, get the BizTalk chm help file, either from a location supplied to you by your course instructor or from Microsoft, <http://www.microsoft.com/en-us/download/details.aspx?id=11503>. Or optionally use the online help if you really do not want to download it locally.

After you have downloaded and unpacked the help file, open it and search for “Using the Unit Testing Feature with Pipelines”.

Open the article and follow instructions (available online at <http://msdn.microsoft.com/en-us/library/dd792682(v=BTS.70).aspx)>.

Exercise 5  
Build and Deploy

In this exercise, you will use the Visual Studio 2015 to build and deploy the project to BizTalk Server 2016.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Build and deploy the project. | 1. In the **Solution Explorer**, select **NWBusinessSolution**. 2. Right click and select **Deploy**. 3. If it is not visible, from the **View** menu, select **Output**. 4. In the Output window, make sure show output from **Build** is selected. 5. Watch the output and verify that the deployment succeeded.   *Notice that the Status bar also displays whether deploy succeeded or failed.* |

Exercise 6  
View deployed assemblies

In this exercise, you will use the BizTalk Server 2016 Administration Console to view the deployed assembly as an application.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. View Resources. | 1. On the **Start** menu, point to **All Programs**, then point to **Microsoft BizTalk Server 2016**, and then click **BizTalk Server Administration**. 2. In BizTalk Server 2016 Administration Console, expand **BizTalk Server 2016 Administration > BizTalk Group > Applications > Lab4**.   *The Administration Console has an asynchronous user interface. When it’s working a small hourglass is visible on the icon in the treeview.*   1. Click on **Resources**. |
| 1. View Resource properties | 1. Notice that the **FKMessaging** and **NWMessaging** assemblies are listed in the right pane. 2. Select **FKMessaging**. 3. Expand the **Name** column. Notice that it displays the Version and the PublicKeyToken. **Lab4**.   *ThePublicKeyToken is a result of the strong name assigned to the project through the use of a key file.* |
| 1. View schema properties | 1. In the left pane, under the **Lab4** application, select Schemas. Notice the **CustomerOrder** and **SupplierPO** schemas.   *If you expand the Assembly column you will notice that it references one of the assemblies located under Resources.* |
| 1. View map properties | 1. In the left pane, under the **Lab4** application, select Maps. Notice the **NWCustomerOrder\_to\_FKSuplier** map.   *If you expand the Assembly column you will notice that it references one of the assemblies located under Resources.* |

Exercise 7  
Deploy a new version

In this exercise, you will use Visual Studio to deploy a new version of a schema and use the BizTalk Server 2016 Administration Console to display the two versions.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Ensure that the assembly was deployed to the Lab4 application. | 1. In **Visual Studio**, return to or re-open the **NWBusinessSolution** solution. 2. Right click the **FKMessaging** project. 3. Select **Properties**. 4. In the left hand pane, select **Application**. 5. Locate the **Assembly Information** button and press it. 6. Locate **Assembly Version**. 7. In the first text box of **Assembly Version**, the Major version, replace the **1** with a **2**.      1. Press ok.   *It is really only necessary to change the Assemmbly Version since the File Versoin is only informational. BizTalk Server doesn’t care if this is changed or not.*   1. Close the **FKMessaging** property pages saving changes. 2. Repeat the above steps for **NWMessaging**. |
| 1. Build and deploy the project. | 1. In the **Solution Explorer**, select **NWBusinessSolution**. 2. Right click and select **Deploy**. 3. If it is not visible, from the **View** menu, select **Output**. 4. In the Output window, make sure show output from **Build** is selected. 5. Watch the output and status bar, and verify that the deployment succeeded. |
| 1. View assemblies | 1. Repeat exercise 6. Notice that there are now two versions of the assemblies deployed.   *If you do not see the two versions, you may have to refresh the* ***BizTalk Administration Console*** *view. Right click the* ***BizTalk Group*** *node and select* ***Refresh****.* |

Exercise 8  
Undeploy assemblies and delete the application

In this exercise, you will use the BizTalk Server 2016 Administration Console to remove the assemblies and then remove the entire application.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Undeploy assemblies. | 1. On the **Start** menu, point to **All Programs**, then point to **Microsoft BizTalk Server 2016**, and then click **BizTalk Server Administration**. 2. In BizTalk Server 2016 Administration Console, expand **BizTalk Server 2016 Administration > BizTalk Group > Applications > Lab4**. 3. Click on **Resources**. 4. Remove the assemblies with version **1.0.0.0** by right clicking them and selecting **Remove**. Start with **NWMessaging**. 5. Navigate to the Schemas and Maps treeview nodes. 6. Notice that the schemas and maps for those assemblies have also been removed. |
| 1. Delete Application | 1. Right click the Application **Lab4**. Select **Delete**. |