# BTSHOL02: Working with Schemas

Objectives

After completing this lab, you will be able to:

Create a new BizTalk project.

Create an XML schema by using the BizTalk Editor.

Promote a schema property.

Create a flat file schema by using the BizTalk Editor.

Validate a schema and generate a sample instance message.

Create a strong name and assign it to an assembly.

Build a schema project.

Scenario

Northwind Traders has chosen to create a custom schema to meet their business requirements for submitting orders to a supplier. In this lab you will create a BizTalk project for the schema, and then define a customer order schema to define the structure of messages (documents) to be exchanged within the solution.

You will also promote several of the schema properties so that the property values can be referenced for content-based routing purposes in the BizTalk Server messaging engine.

Estimated time to complete this lab: 45 minutes

After you have created the schemas and promoted the schema properties, you will use the BizTalk Editor to generate a sample instance message and validate the schemas to check for any inconsistencies.

User Name: **Administrator**

Password: **pass@word1**

Exercise 1  
Creating a New BizTalk Project

In this exercise, you will build a new solution for this project by using the Microsoft Visual Studio® 2015 environment. First you will create a new blank solution, and then you will add a new project to the solution. This project will contain the schemas that you will create in the following exercises.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create a blank solution.  A solution might include only one BizTalk project if the solution is relatively simple.  A solution might include many BizTalk projects if the solution consists of numerous projects that are developed independently and then must be integrated later. | 1. Click Start, click Microsoft Visual Studio 2015.   The Visual Studio 2015 splash screen appears and then Visual Studio starts.   1. In Visual Studio, from the **File** menu, click **New**, and then click **Project**. 2. In the New Project dialog box, expand **Other Project Types**, click **Visual Studio Solutions**,and then click the Blank Solution template. 3. In the **Name** box, type **NWBusinessSolution** as the solution name. 4. In the **Location** box, type **C:\Labs\Lab 2\Start**. 5. ClickOK to open the new solution.   A new solution appears in Solution Explorer and the solution files are created in C:\Labs\Lab 2\Start\NWBusinessSolution. |
| 1. Add a new BizTalk project to the solution.  Selecting the BizTalkproject template exposes the BizTalk tools, such as BizTalk Mapper, within the Visual Studio .NET development environment, and invokes BizTalk‑specific build tasks. | 1. In Solution Explorer, right-click Solution '**NWBusinessSolution**', point to Add, and then click New Project. 2. In the Add New Project dialog box, in the Project Types pane, click **BizTalk Projects**. 3. In the Templates pane, select Empty BizTalk Server Project. 4. In the **Name** box, type **NWMessaging**, and then click **OK**.   A new project is added to Solution Explorer, and the starting project files are created in C:\Labs\Lab 2\Start\NWBusinesSolution\NWMessaging.   1. In Solution Explorer, under **NWMessaging**, expand **References** to view the default references that get added to a new project. |

Exercise 2  
Creating an XML Schema Using BizTalk Editor

In this exercise, you will create a schema to define the structure of the shipping address record. You will also open an existing schema and use the **import** function to insert schema nodes within the existing schema that use the newly created address definition.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create the shipping address schema.             Selecting the Schema template causes BizTalk Editor to start. | 1. In Solution Explorer, right-click the NWMessaging project, point to Add, and then click New Item. 2. In the Add New Item dialog box, in the **Categories** pane, select **Schema Files**, and then in the **Templates** pane, select **Schema**. 3. In the Name box, type ShippingAddress.xsd, to name the schema.      1. Click Add to open the blank schema in BizTalk Editor.   The schema tree (left pane) and XSD view (right pane) appear in BizTalk Editor. Also, the new schema (ShippingAddress.xsd) is added to Solution Explorer. |

(*continued*)

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create the schema record.   The **Root** node should always be renamed using a meaningful name that represents the type of document described by the schema.    You can right-click an empty area of the Toolbar and select **BizTalk Editor** to display a floating toolbar of BizTalk Editor options. | 1. In the Schema Tree, right-click the Root node, and then click Rename. 2. Type ShippingAddress as the new name, and then press ENTER. 3. Right-click the ShippingAddress node, point to Insert Schema Node, and then click Child Field Element to add the following fields (repeat this step for each field element):  * City * Country * Street * Email * Fax * State * Zip * Phone   The completed schema should look like this:  Lab02-4   1. On the File menu, click Save All. |
| 1. Open an existing schema and import a new child record.  This allows you to open an existing schema and use the **import** function to insert schema nodes within an existing schema allowing you to reuse schema types.  You can show properties in alphabetical order by clicking the **Alphabetic** button at the top of the Properties window. | 1. In Solution Explorer, right-click the NWMessaging project, point to Add, and then click **Existing Item**. 2. Browse to **C:\Labs\Lab 2\Start**, and then double-click the **CustomerOrder.xsd** schema to add it to the project.   Theschema is added to Solution Explorer.   1. In Solution Explorer, double-click **CustomerOrder.xsd** to open it in BizTalk Editor. 2. If the **Properties** window is not visible, then in the BizTalk Editor, right-click the **<Schema>** node of **CustomerOrder.xsd**, and then select **Properties**. 3. In the **Properties** window, click the **Imports** property, and then click the **Ellipsis (…)** button. 4. In the **Imports** dialog box, verify that **XSD Import** is selected from the drop-down box, and then click **Add**. 5. In the **BizTalk Type Picker** dialog box, expand **NWMessaging**, expand **Schemas**, click **NWMessaging.ShippingAddress**, and then click **OK**.   Untitled   1. Click **OK** to close the **Imports** dialog box. 2. Right-click the **CustomerOrder** root node, click **Insert Schema Node**, and then click **Child Record**. 3. Click the child record **Record**, then in the **Properties** window, select the **Data Structure Type** property,andfrom the list, select **ns1:ShippingAddress (Reference)**.   TheShippingAddressrecord is added to the schema and includes all of the child elements you defined in the first exercise.   1. On the **File** menu, click **Save All**. |

Exercise 3  
Promoting Schema Properties

In this exercise, you will promote schema properties so that the property values can be referenced by BizTalk messaging and orchestration. Promotion is a mechanism that you must use when you want to reference a specific value within a message instance and then use the value to make message routing decisions. Additionally, the promoted property will be visible by IntelliSense in Expression Editor when code is required in an orchestration.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Promote property fields.   The property fields are promoted so that their values can be referenced for content-based routing purposes and orchestration.  Property schema define the common name for promoted properties allowing you to promote values from different schemas as the same logical property. | 1. In BizTalk Editor, in the Schema Tree, right-click the **<Schema>** node, and then click **Expand Schema Node** to show the entire schema. 2. Under the **Customer** **Order** node, right-click CustomerPONumber, point to Promote, and then click Quick Promotion. 3. When prompted, click **OK** to add the property schema to the project.   A new schema (PropertySchema.xsd) is added to the Solution Explorer and is opened in the BizTalk Editor.   1. Close the PropertySchema.xsd file to return to editing the CustomerOrder.xsd file.   The icon for the CustomerPONumber has changed to indicate that the field has been promoted.   1. Right-click TotalOrderAmount, point to Promote, and then click Quick Promotion. |
| 1. Promote distinguished fields.  The distinguished fields are promoted so that their values can be referenced in an orchestration that you will create in later labs.   Distinguished fields cannot be used to make routing decisions. | 1. Lab02-5Right-click the <Schema> node, point to Promote, and then click Show Promotions. 2. Select the following fields from the Schema (left) pane, and add them as distinguished fields by clicking the Add >> button to move them to the list of distinguished fields on the right side of the dialog:  * Status * Comments  1. Click **OK**.   The completed schema should look like this:   1. On the **File** menu, click **Save All**. |

Exercise 4  
Creating a Flat File Schema using The Flat File Schema Wizard

In this exercise, you will create a flat file schema to define the structure of a customer contact record. Flat file schemas provide the translation between the flat file format, such as comma separated values (CSV) and XML. BizTalk uses these schemas to parse flat file documents into their related XML structure.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create the customer contact schema.   This creates the XSD schema that describes the XML structure that flat files will be converted to.   The schema will contain metadata providing details on the flat file format including delimiters, wrap characters, field sizes, etc. | 1. In Solution Explorer, right-click the NWMessaging project, point to Add, and then click New Item. 2. In the Add New Item dialog box, in the Templates pane, select **Flat File Schema Wizard**. 3. In the Name box, type CustomerOrderFlatFile.xsd, and then click **Add**. 4. In the **BizTalk Flat File Schema Wizard**, click **Next**. 5. On the **Flat File Schema Information** page, in the **Instance file** box, browse to or type **C:\Labs\Lab 2\Start\CustomerOrder.txt.** 6. In the **Record name** box, type **Customers**, and then click **Next.**   Untitled   1. On the **Select Document Data** page, leave all the data selected, and then click **Next**. 2. On the **Select Record Format** page, verify that **By delimiter symbol** is checked, and then click **Next**. 3. On the **Delimited Record** page, set the **Child Delimiter** to {CR}{LF}, and then click **Next**. 4. On the **Child Elements** page enter **“Customer”** for the **Element Name** of the first item (Customers\_Child1). 5. Change the **Element type** of the first item to **Repeating record**. 6. Change the **Element type** of the second item to **Ignore**.   Untitled   1. Click **Next** to finish defining the root record.   You will be returned to the beginning of the wizard, this time to define the child record format.   1. Select **Next** to begin defining the Customer record. 2. On the **Select Document Data** page, ensure that only the first line of text is selected and click **Next**.   Untitled   1. On the **Select Record Format** page, verify that **By delimiter symbol** is checked, and then click **Next**. 2. On the **Delimited Record** page, set the **Child Delimiter** to **,** (Comma), and then click **Next**. 3. On the **Child Elements** page, change the **Element Names** to:  * **FirstName** * **MiddleName** * **LastName** * **ContactNo**   Untitled   1. Click **Next,** and then click **Finish** to create the schema. |

Exercise 5  
Validating Schemas and Generating Instance Messages

In this exercise, you will use the **Validate Schema** command to determine whether the schemas contain any internal inconsistencies, or have other issues that might prevent them from being used effectively for processing instance messages. You will also generate a sample instance message for each of the schemas.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Validate the schema file.  Validating the schema ensures that an XML document can be created that would be well-formed and valid against this particular schema. | 1. In Solution Explorer, right-click CustomerOrder.xsd, and then click **Validate Schema**.   The results of the schema validation are displayed in the output window. The schema validation tool is helpful when using schemas you’ve received from other parties or trading partners, that may not have been generated using BizTalk Server. BizTalk schemas will always validate. |
| 1. Generate a sample instance message for the schemas.  Generating a sample instance provides a visual test as well as a document to pass through the process for testing. | 1. Select the **CustomerOrder.xsd** file in Solution Explorer 2. In the Properties window,  type **C:\Labs\Work\Lab 2\GeneratedCustomerOrder.xml** in the **Output Instance Filename** box, and press ENTER. 3. In Solution Explorer, right-click **CustomerOrder.xsd**, and then click **Generate Instance**.   A sample instance message is saved in C:\Labs\Work\Lab 2\, and a link to the XML instance is shown in the Output window.   1. Click the link while pressing CTRL to view the resulting XML file. |
| 1. Validate the sample instance message.  The CustomerOrder.xml XML file is provided so that you can check for errors in the schema you just created. | 1. In Solution Explorer, click **CustomerOrder.xsd** to highlight it. 2. In the Properties window,  type **C:\Labs\Lab 2\Start\CustomerOrder.xml** in the **Input Instance Filename** box, and then press ENTER. 3. Right-click the **CustomerOrder.xsd** schema file, and then click **Validate Instance**.   The results of the instance validation are displayed in the output window. This step validates the schema against an actual XML file. |
| 1. Validate the flat file sample instance.  The CustomerOrder.txt flat file is provided so that you can check for errors in the schema you just created. | 1. In Solution Explorer, click **CustomerOrderFlatFile.xsd** to highlight it.  |  |  | | --- | --- | | Property | Value | | **Input Instance Filename** | C:\Labs\Lab 2\Start\CustomerOrder.txt | | **Validate Instance Input Type** | Native |  1. In Properties window, verify the following properties: 2. Right-click the **CustomerOrderFlatFile.xsd** schema file, and then click **Validate Instance.**   The results of the instance validation are displayed in the output window. This step validates the schema against an actual flat file.   1. In the **Output** window, while holding CTRL click the top link (ends with **output.xml**).   The window that opens displays the xml that was generated by parsing the flat file. Notice that there are two customer elements, one for each line in the original file.   1. Close the preview pane. |

Exercise 6  
Sign and Build the Project Output

In this exercise, you will create and assign a strong name for the BizTalk assembly that you are about to build. A strongly-named assembly provides several security benefits and allows for side-by-side versioning of assemblies. A strong name guarantees the uniqueness of the assembly by assigning a digital signature and a unique key pair.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create a strong name assembly key file.  This creates and assigns the key file to the assembly. You must do this for each of the projects in this solution. | 1. In Solution Explorer, right-click the **NWMessaging** project and select **Properties**. 2. Select the **signing** tab to display the signing property page. 3. Check the box labeled **Sign the assembly**. 4. In the drop down box, select **<New. . . >** 5. When prompted enter **“Key.snk”** for the Key file name and uncheck the box to protect the key file with a password.   Untitled   1. Choose **Save all files** from the **File** menu. |
| 1. Build the project.  The assembly is compiled into a DLL file and saved in the *\bin\Debug* folder within the project folder. | 1. In Solution Explorer, right-click NWMessaging, and then click Build.   The results of the compiled schema project are displayed in the output window.   1. Close Visual Studio. |

Optional  
Creating better contracts by specifying types and restrictions

In this optional exercise, you will restrict the content of the schema you created to follow certain restrictions to make it an even more specific contract of the expected format of the message.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create a type restriction | 1. In the **NWMessaging** project, and open the ShippingAddress schema. 2. Locate and select the **Email** element. 3. In the **Properties** window, set **Base Data Type** to *xs:string* 4. Set **Derived By** to *Restriction* 5. Now use the **Pattern** property to set a restriction of *^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$* 6. Build the solution. |
| 1. Generate a XML instance | 1. Using the **ShippingAddress** schema, Right-click it and **Generate Instance**. 2. In the **Output** pane, hold ctrl and click the link. 3. In the browser window, right click and select **View Source**. 4. In the XML editor that opens, locate the XML Toolbar and click the **Reformat** button. Untitled 5. Go to the File menu, and save the XML document to the C:\Labs\Work\Lab 2 with a name of your choosing. |
| 1. Validate instance | 1. In the **Properties** pane of the **CustomerOrder** schema, go to the **Input Instance Filename** property and locate the file you just created. 2. Right-click the schema and select **Validate Instance** *At this point you will get an error.* 3. Correct the **Email** field in the XML instance that you have open in your XML editor windows so that it matches the constraint by entering a **valid email address**. 4. **Validate instance** (which should succeed) |