# BTSHOL14B: BizTalk Server – B2B

Overview

In this lab, you will learn how to utilize the B2B capabilities shipped with BizTalk Server 2010. After completing this lab, you will be able to:

* Define partners and partner profiles in the Trading Partner Management tool (TPM)
* Define Agreements between partners
* Configure Send- and Receive ports with EDI Pipelines
* Monitor EDI interchanges in the Administration Console.

Scenario

The Sales department within you organization is using Axapta, and needs to send invoices to a trading partner (Contoso). After the order has been transmitted, you will monitor the interchange for incoming CONTRL verifications.

Estimated time to complete this lab: 45 minutes

User Name: **Administrator**

Password: **pass@word1**

Exercise 1

Familiarize yourself with the solution

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Open the solution, and examine the its content | 1. Open the following solution: *C:\Labs\Lab 14B\Start\ Lab14B.sln* 2. The Lab14B.InvoiceProcess project has two schemas, one transformation and one pipeline. |
| 1. Examine the schemas | 1. In the Solution explorer, open *EDIPLUS\_Invoic\_AL014\_v2.xsd*. This schema represents the flat-file we’ll receive from the Sales Department (Axapta). 2. Open the *EFACT\_D93A\_INVOIC.xsd* schema. This is the outgoing EDIFACT message sent to the trading partner (Contoso).   As you can see, these schemas are pretty big. EDI/EDIFACT schemas are often used for many different branches and organizations, and are therefore quite generic. |
| 1. Examine the map | 1. In the Solution Explorer, open *EDIPLUS\_Invoic\_to\_D93AINVOIC.btm*.   As the schemas are big the mappings become complex. It’s therefore a good practice to separate different sections of the map into different pages.  C:\Users\ADMINI~1\AppData\Local\Temp\SNAGHTML5d1a0ee.PNG  A page might seem to be empty because the functoids are not visible on the design surface. This is because the design surface is much bigger then then the visible area.  By using the Grid Preview you can easily find the functoids. Right-click the design surface and select “Grid Preview”. |

Exercise 2

Deploy solution and import bindings

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Deploy the project to BizTalk and import the bindings. | 1. Right-click the Lab14B.InvoiceProcess project and select deploy. This will create an Application in BizTalk called *Lab14B*. 2. Open BizTalk Administration Console. 3. Right-click the Lab14B Application, and select *Properties*. 4. Click *References* in the left pane, and add the “BizTalk EDI Application”. Click *Ok* to close the dialog.   Your EDI solution uses the EDI pipelines which are deployed to the BizTalk EDI Application, and must therefore be referenced.   1. Browse to the Lab14B Application. Right-click the Application and select *Import -> Bindings.* 2. Browse to *C:\Labs\Lab14B* and select *Lab14B.BindingInfo.xml.*   This will create two Receive- and one Send port.   |  |  | | --- | --- | | Port name | Purpose | | RcvInvoice\_SalesDep | Receives flat file invoices from Axapta | | SndInvoice\_Contoso | Sends EDI/EDIFACT invoice to trading partner. | | RcvCONTRL\_Contoso | Receives control message from trading parner. | |

Exercise 3

Create Parties, Profiles and Agreements

Agreements are evaluated at runtime by Receive- and Send ports using the EDI Pipelines. In this case you are sending an EDIFACT message to Contoso, so you’ll be using the *EDI Send* pipeline.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Create Parties | 1. In the BizTalk Administration Console, right-click *Parties* and select *New->Party*.   C:\Users\ADMINI~1\AppData\Local\Temp\SNAGHTMLb5ad3ec.PNG   1. Set the name to “**SalesDepartment – Internal**”. Leave the checkbox “Local BizTalk processes…” selected.   C:\Users\ADMINI~1\AppData\Local\Temp\SNAGHTML38d78a.PNG  This checkbox indicate that this is an internal party. In turn, this means BizTalk is responsible for things like managing interchange numbers, batching etc.   1. Click *Ok to close the window.* 2. In the *Parties and Business* designer, expand the *SalesDepartment – Internal* party and double-click on the *SalesDepartment - Internal\_Profile*. 3. In the *Properties* box on the left hand side, select *Identities*.   Each party has its unique identity. There are different types of identities, and they are made up by a Qualifier and the actual id. As there are quite a few different qualifiers in the name list, it is easier to type the Qualifier first, instead of selecting if from the list.   1. Select the *Qualifier* column and type “14”. Select *EAN (European Article Numbering Association)* in the *Name* column. Set the value to “123”.   A party can have any number of profiles. A Profile acts as a template, when setting up the agreement.   1. At the top of the *Profile – Properties* dialog, select *Add protocol settings -> Encoding settings -> EDIFACT:*   C:\Users\ADMINI~1\AppData\Local\Temp\SNAGHTMLb6d32ce.PNG   1. This will create an “EDIFACT\_Settings\_1” tab. Leave all settings and click *Ok.* 2. Repeat all the steps above, to create the Contoso party:  * Party name = Contoso * Unselect the “Local BizTalk processes…” checkbox, as this is an external party. * Open the *Contoso\_Profile* profile and set the *Qualifier* to “14” and the identity to “456” |
| 1. Create Agreement | 1. In the *Parties and Business Profiles* designer right-click the *SalesDepartment\_Profile* and select *New->Agreement.* 2. Change the name to “SalesDep\_Contoso Agreement” 3. Set the protocol to *EDIFACT*, and select “Contoso” as the second party. 4. Check the ***Turn ON*** *reporting* checkbox. 5. Click *Apply* to save the agreement.   As you might recall from the Scenario description, the SalesDepartment requires an acknowledgement from Contoso. We could have indicated this on the Profile, but we’ve chosen to set this on the Agreement instead.   1. Click the *SalesDepartment-Internal->Contoso* tab. 2. Select *Acknoledgements* in the left pane, and mark both checkboxes in the *EDIFACT ACK(CONTRL)* group as shown below:   C:\Users\ADMINI~1\AppData\Local\Temp\SNAGHTMLb9d3a10.PNG   1. Select *Send Ports* in the left pane, and add the existing “SndInvoice\_Contoso” port created earlier.   When a message is passing the EDI Send pipeline in the SndInvoice send port, the pipeline will fetch the Agreement information to create an envelope (UNB segment).  To complete this step, you’ll need to add the send port to the Contoso Party.   1. Click *Ok* to close the agreement window. 2. Open the Contoso party, by double clicking it in the *Parties and Business Profiles*. Make sure the “SndInvoice\_Contoso” is added to the list of send ports. |

Exercise 4

Start and test the application

You will now test the solution by sending a flat file invoice from the Sales Department to Contoso, which in turn will receive an EDIFACT INVOIC message. Upon receiving the invoice, Contoso will send back an acknowledgement to BizTalk.

|  |  |
| --- | --- |
| Tasks | Detailed steps |
| 1. Start the Application | 1. In the BizTalk Administration Console, right-click the Lab14 application and select *Start*. This will also start the *BizTalk EDI Application*. 2. Open *C:\Labs\Work\Lab 14B* and copy INVOIC.txt to the *FromSalesDepartment* folder.   The flat file is going to be disassembled through the RcvEDIPLU\_Receive pipeline and then mapped to the EFACT\_D93A\_INVOIC format.   1. Open the *ToContoso* folder, and verify the outgoing message to Contoso.   After Contoso receives and validates the message, they are expected to send an acknowledgement back to BizTalk.   1. In BizTalk Administration Console, go to the *Group Hub* by selecting the *BizTalk Group* node in the top of the tree view. At the bottom of the Group Hub you’ll find the *EDI Status Reports*. Click the *EDI Interchange and Correlated ACK Status* report. The status report should have an interchange with Control Id (Interchange Id) equals 1 with a status set to *Ack Expected*. 2. In the *C:\Labs\Work\Lab 14B* folder, open the CONTRL.txt message. The first element in the **UCI** segment is the *InterchangeId*:   UCI+**1**+456:14+123:14+8'  If you run the sample again, you need to increment the Interchange id to correlate to the one created for the outgoing exchange.  As there are no send port or orchestration subscribing to the incoming CONTRL message, you will get a routing exception. For the purpose of this lab, just disregard this suspended message. |