IBM Integration Bus
V10.0

Lab 1

Create, Deploy and Test
Your 1st Integration Service

V1.0 January 2015

Table of Contents

Overview

Description

In the lab you will create your 1st IIB Integration Service.

You will use a sample WSDL file to expose an IIB Integration Service as a web service.

This simple service will log customer address details to file and return a customer address object enriched with city and country data.

Pre-requisites

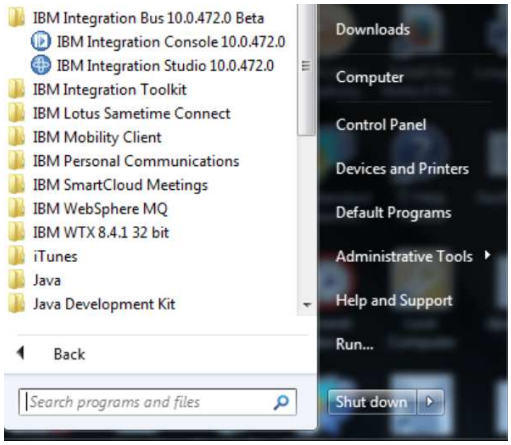
You must have downloaded and installed (unpacked) the IIB V10 open beta or IIB V10 developer edition product. See the previous tutorials on the Getting Started with IIB web site.

The DemoCustomerBasic.wsdl file from the Getting Started with IIB web site

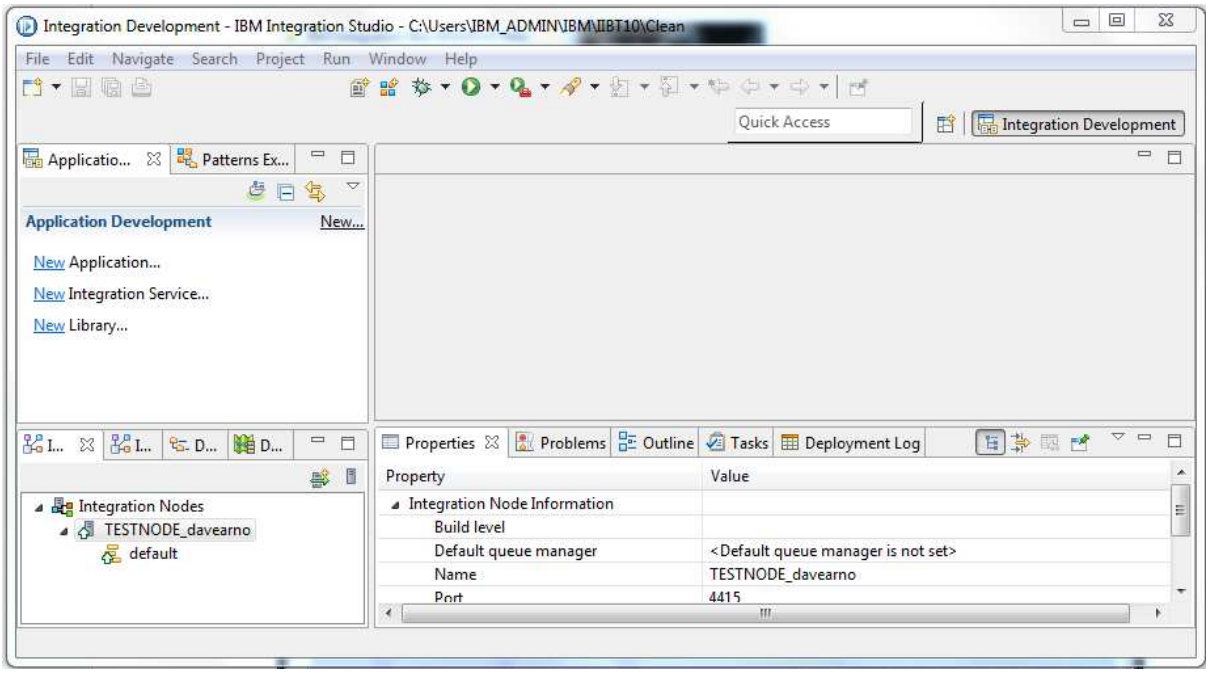
Instructions

Constructing the IIB Integration Service in the IIB Studio

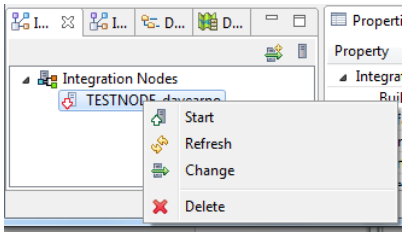
1. Open the IBM Integration Studio and select a new workspace.



Click on Go to the IBM Integration Studio to close the welcome screen and you will be presented with the development and testing workspace



The clean workspace will have a TESTNODE configured and ready to use, see the Integration Nodes pane in the bottom left hand corner. If the TESTNODE is not started right click on it and select Start



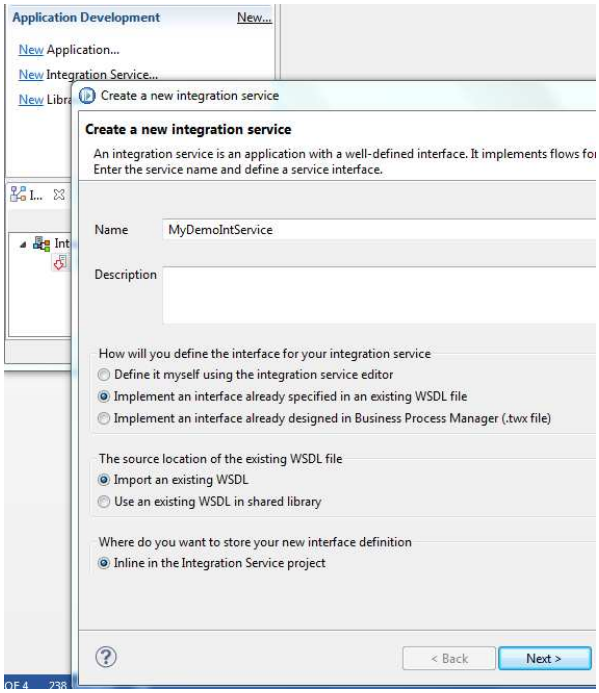
2. Start by using the New Integration Service wizard to import your WSDL file and generate the integration service framework (skeleton artefacts).

Give the service a name.

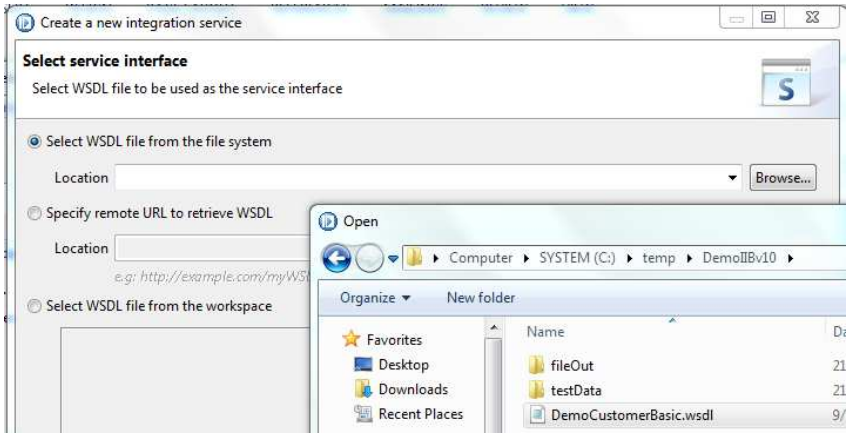
Select “Implement an interface already defined in a WSDL file”

Select “Import an existing WSDL file”

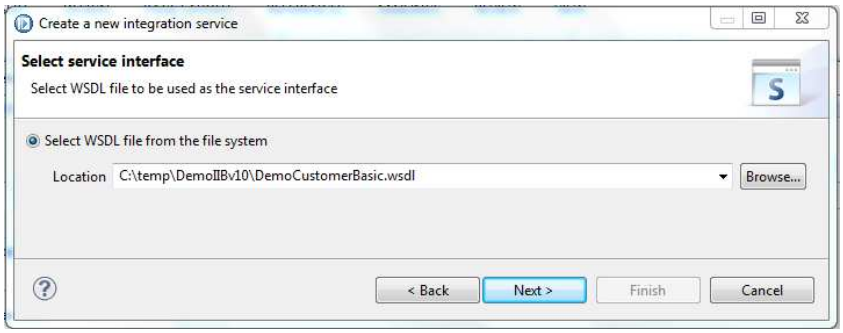
Let the remaining radio buttons default as shown below then click Next



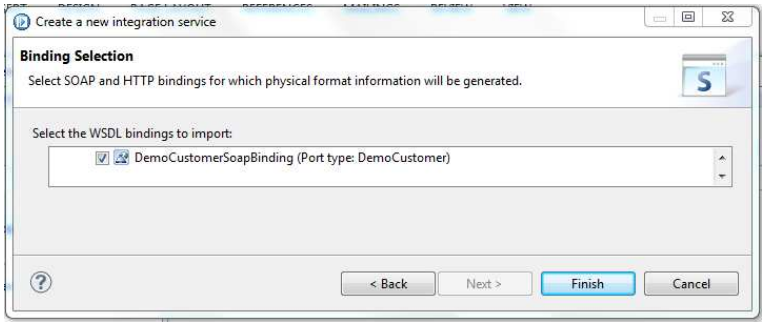
Select the sample WSDL file DemoCustomerBasic.wsdl by clicking Browse and then opening the file.



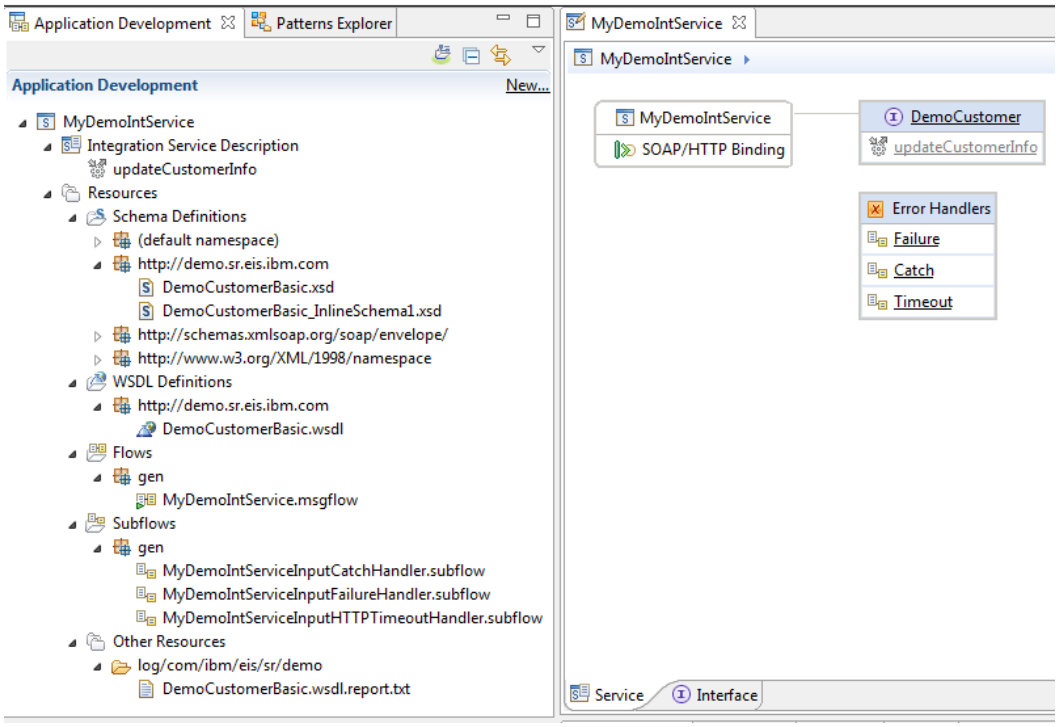
Click Next



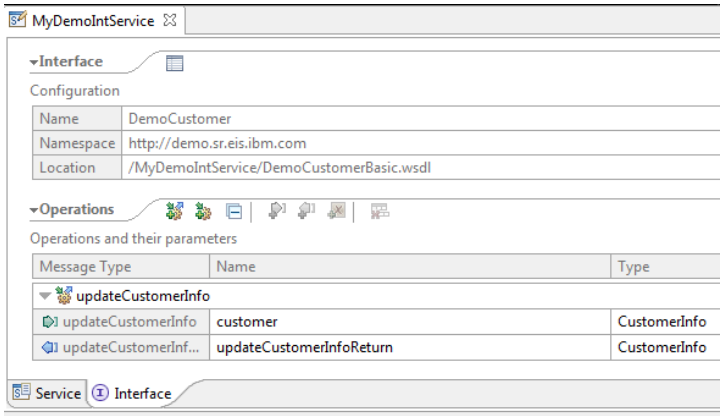
Select the binding to import (there is only 1) then click Finish and the IIB Studio will build the skeleton service for you.



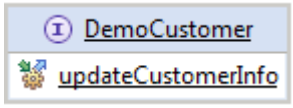
Explore the generated artefacts – the MyDemoIntService Integration Service Description is open in the right hand pane in the Service view.



Click on Interface to review the interface definition

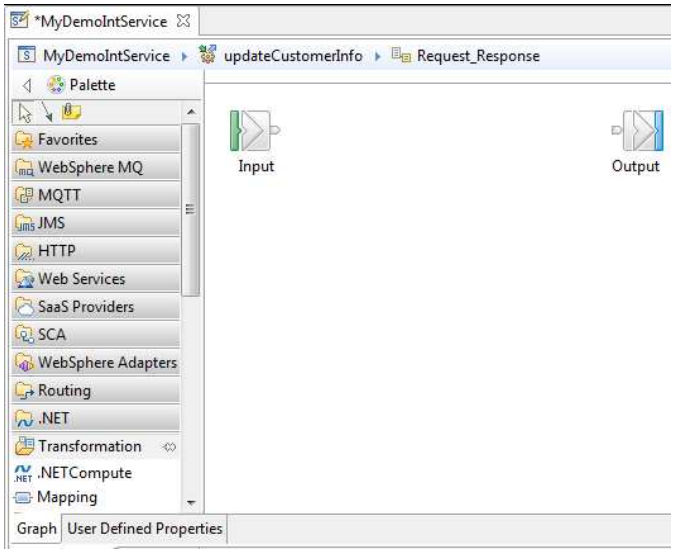


3. Build the integration flow behind the service façade

Returning to the Service view of the MyDemoIntService Integration Service Description click on the updateCustomerInfo link 

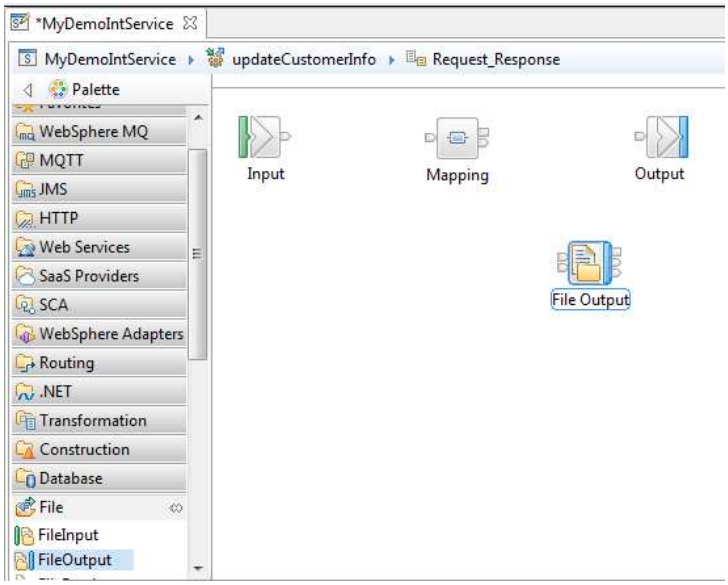
to open an IIB Message Flow editor in which you will build out the functionality for this service.

A Request_Response subflow is generated and you place your IIB functional nodes between the Input and Output terminals.



Select a Mapping node from the Transformation folder in the palette (we will use this to build the updateCustomerInfoResponse based on the updateCustomerInfo request data plus some hard coded values) and place it between the Input and Output terminals.

Select a FileOutput from the File folder (we will use this to log our customer info data to file) and place it below the Output terminal

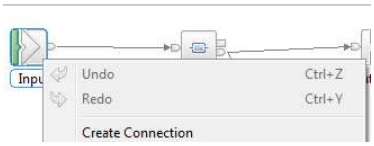


Wire the terminals and nodes together as follows:

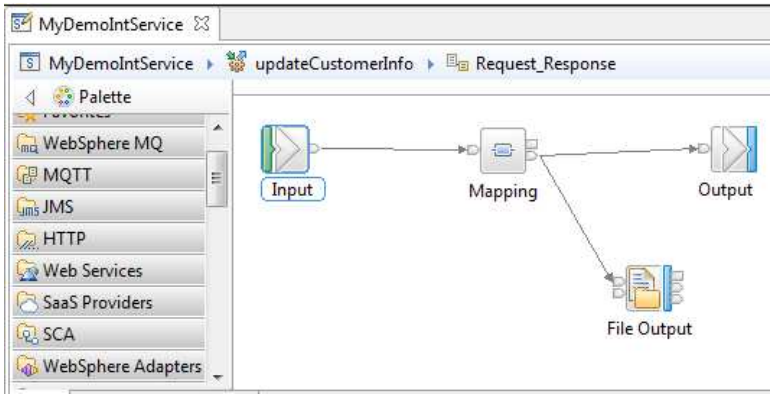
a) Input Node (Out terminal) to Mapping Node (In terminal)

b) Mapping node (Out terminal) to Output Node (In terminal)

c) Mapping node (Out terminal) to File Output Node (In terminal)

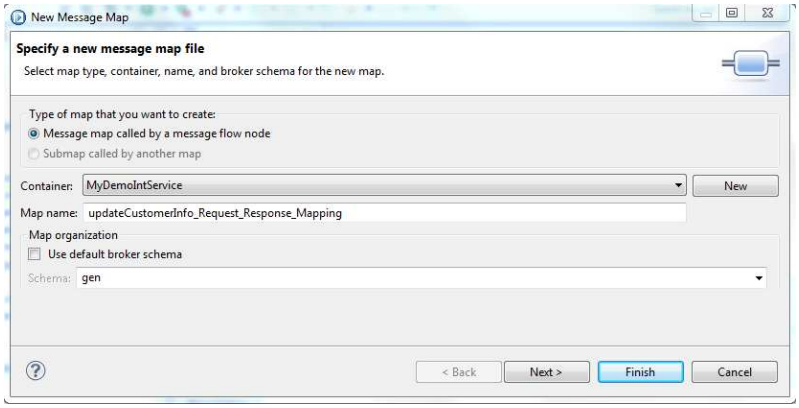
You can hover over the terminals with the mouse or right click on a node and select create connection 

The result should be as show below



4. Create the mapping to map from the Customer Request object to the Customer Response object and hard code values for city and country.

Double click on the Mapping node to open the map creation wizard, accept the defaults and click next

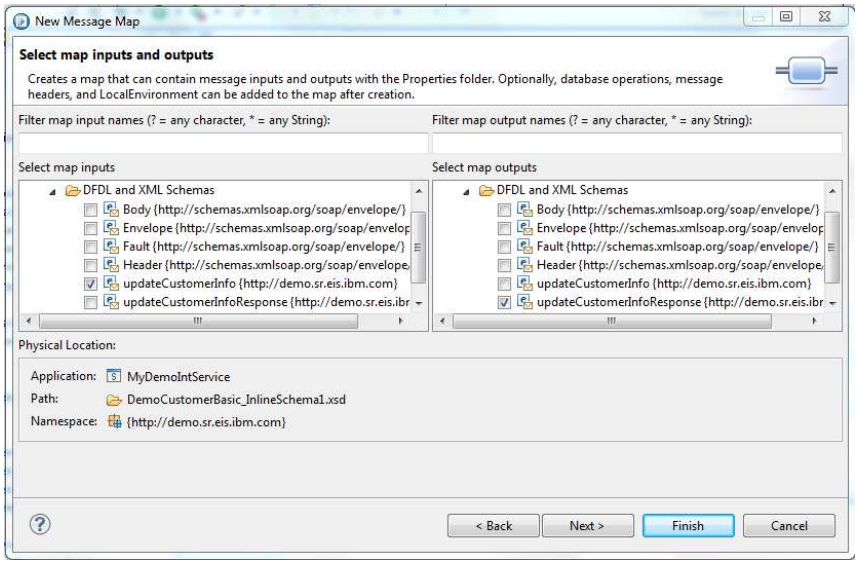


Select the map Inputs and Outputs.

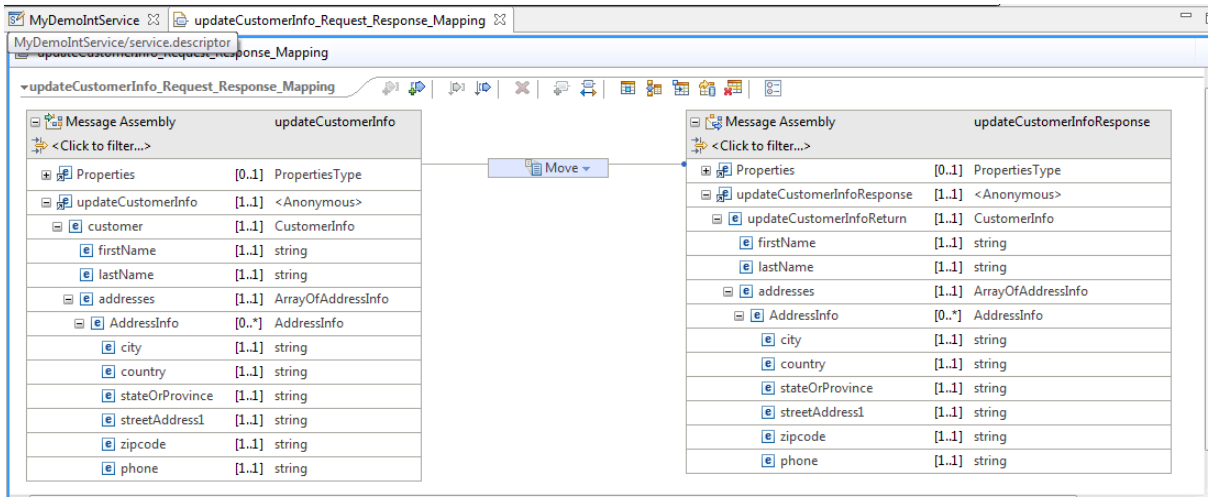
Expand the service name node then expand DFDL and XML Schemas node in each side of the mapping wizard window

Map Input = updateCustomerInfo, Map Output = updateCustomerInfoResponse

Then click finish

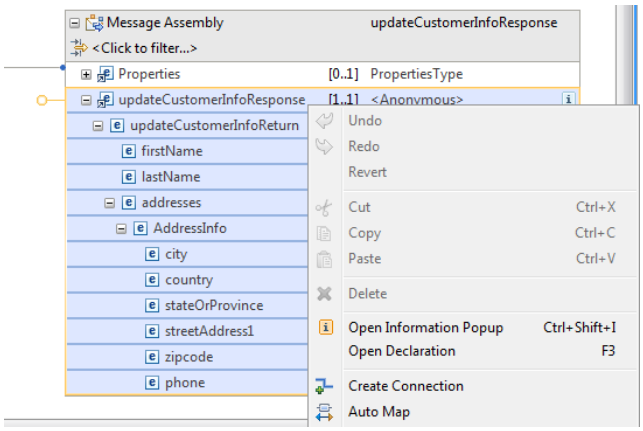


In the mapping editor, expand the input and output elements to the review the structures

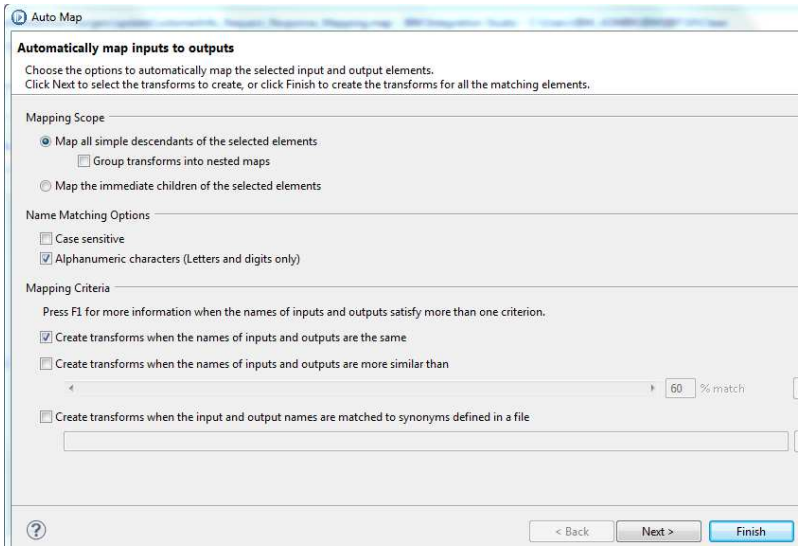


Use Auto Map to do the hard work for us.

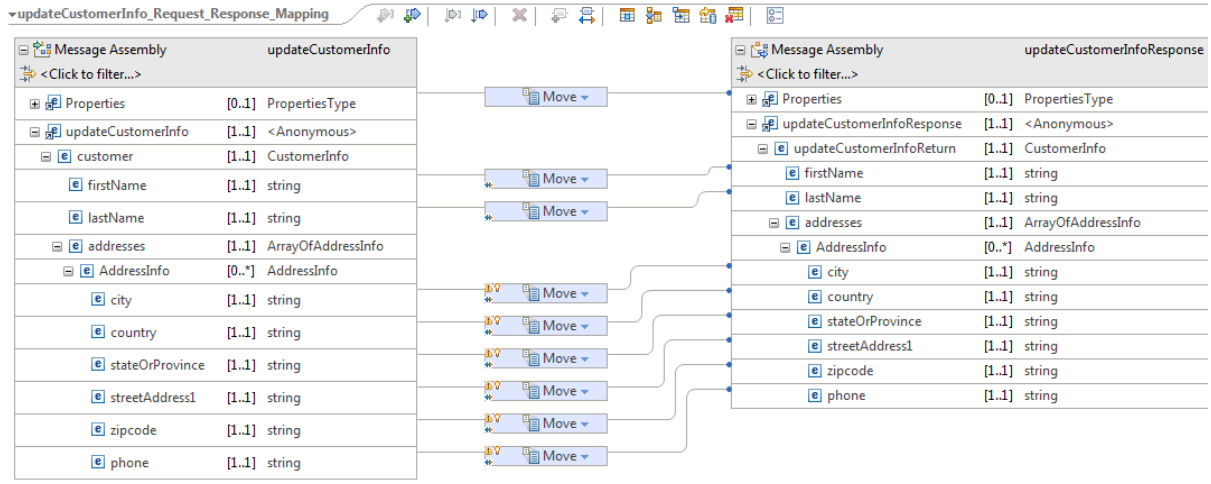
Right click on updateCustomerInfoResponse and select Auto Map.



Accept the defaults and click Finish

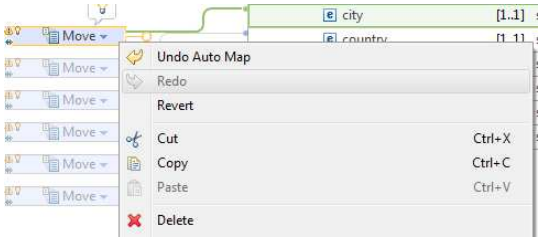


The result should be as follows

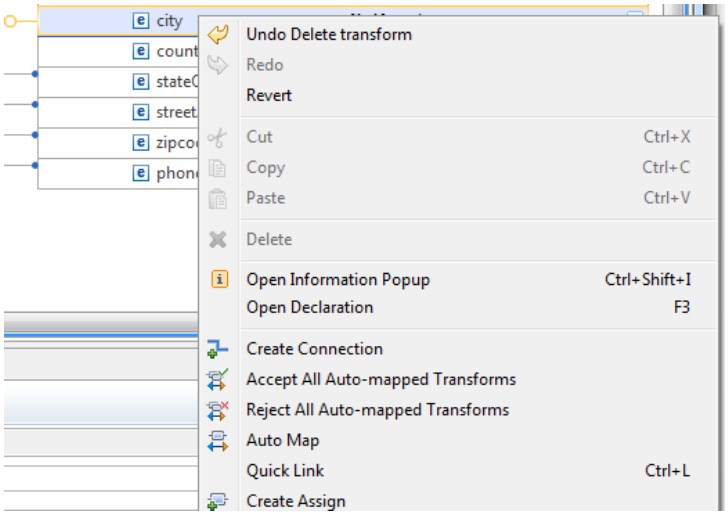


We will be hard coding the city and country so delete the Move “boxes” associated with these elements and create assigns for them.

Right click on the Move box for city and select delete, then repeat for country

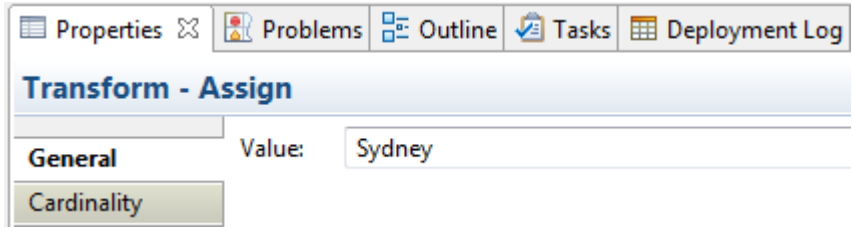


Right click on the city element on the output side and select Add Assign, then set a value

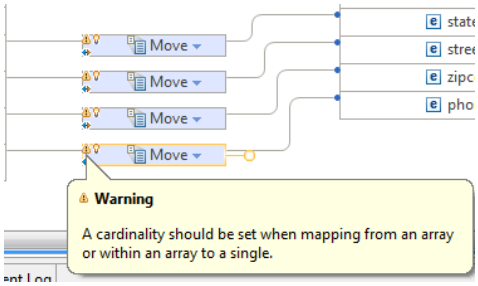


Fill in a city name in the properties.

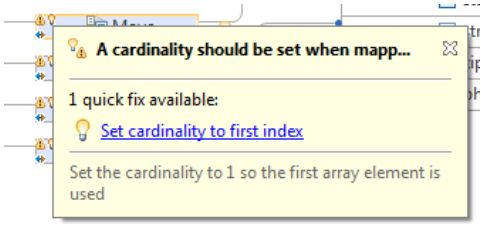
Repeat this for the country element

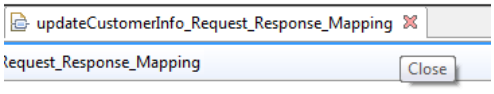


Finally, because the AddressInfo object is an array we have warnings that out Move statements do not have a cardinality. You can hover the mouse over the yellow warning sign to read the message.



Use QuickFix to set a cardinality. Hover the mouse over the Lightbulb icon on the Move box for the state element. The click on the Set cardinality to first index link to “fix” the problem. Repeat for the other three elements.



Hit Ctrl-S to save the completed map. Close the map 

5. Configure the File Output node to log a record of the output customer info data to file.

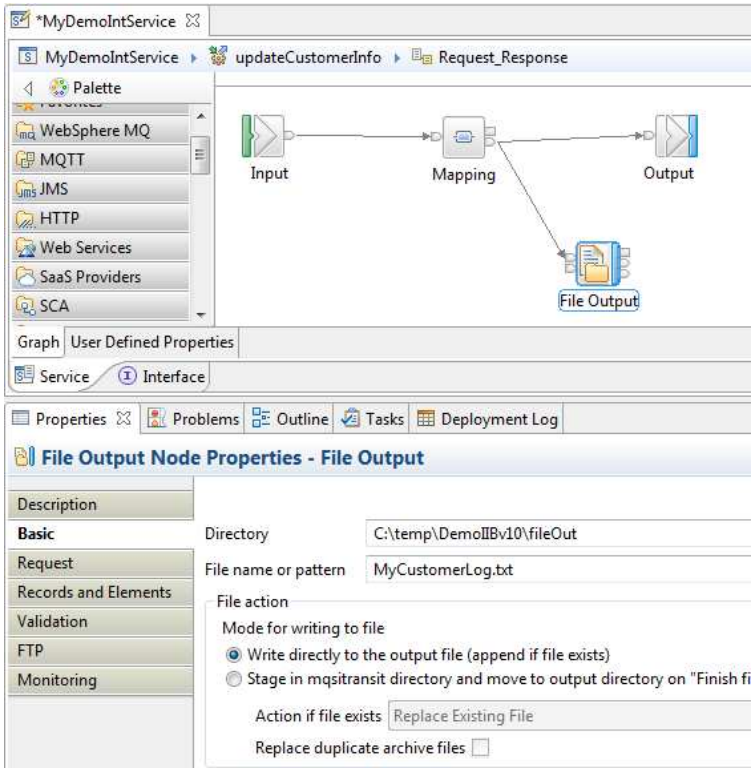
Click on the File Output Node and configure its properties.

Set a Directory for the file to reside in. Make sure this directory already exists, or create the directory in your file system.

Set the File name.

Select radio button for “Write directly to the output file(append if file exists)”

All other properties on all tabs can be left as default.



Hit Ctrl-S to save MyDemoIntService. The development exercise is now completed.

Deploying and testing the IIB Integration Service.

The IIB Studio has an integrated test client that can be used to:

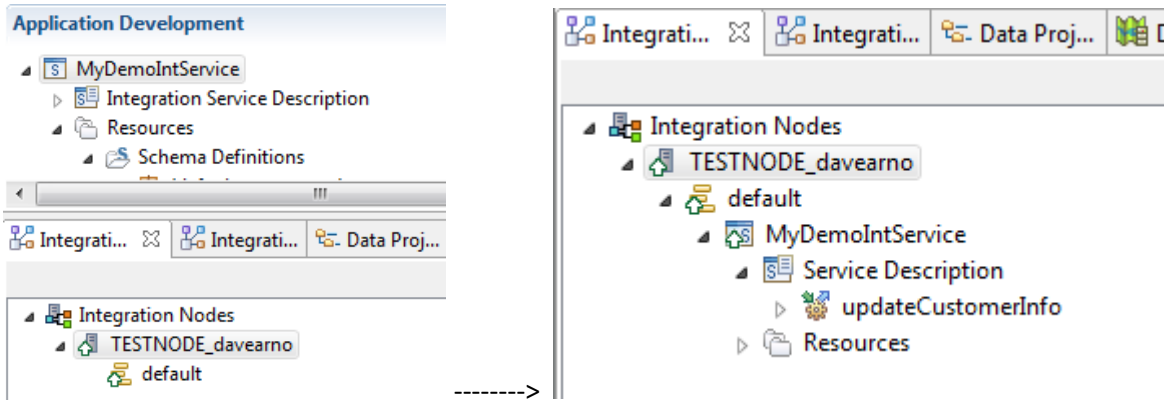
a) Deploy
b) Create test data
c) Invoke the Integration Service with the test data
d) Capture response data.

Before we use the test client, for demonstration purposes only, the next couple of steps show you how to manually deploy and remove the Integration Service from the runtime.

Deploying to the runtime.

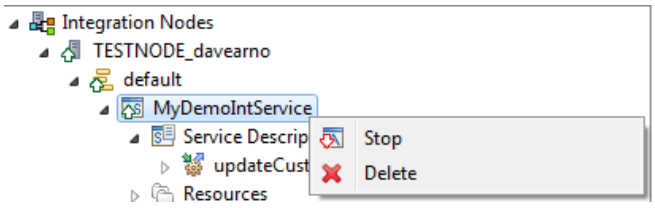
Expand the TESTNODE_name Node object to reveal the default integration server object

Click on the MyDemoIntService and drag and drop it onto the TESTNODE_*name*->default integration server.



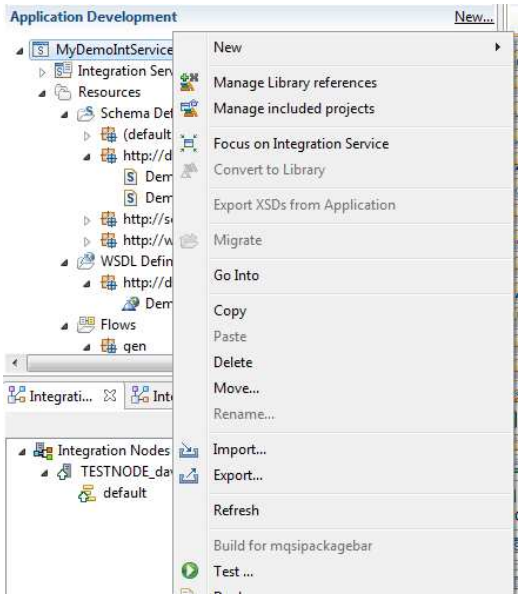
Removing artefacts from the runtime.

Click on the MyDemoIntService under the TESTNODE_*name*->default integration server and select delete.



Deploying and testing using the Test Client.

Right click on MyDemoIntService in the top left of the IIB Studio and select Test

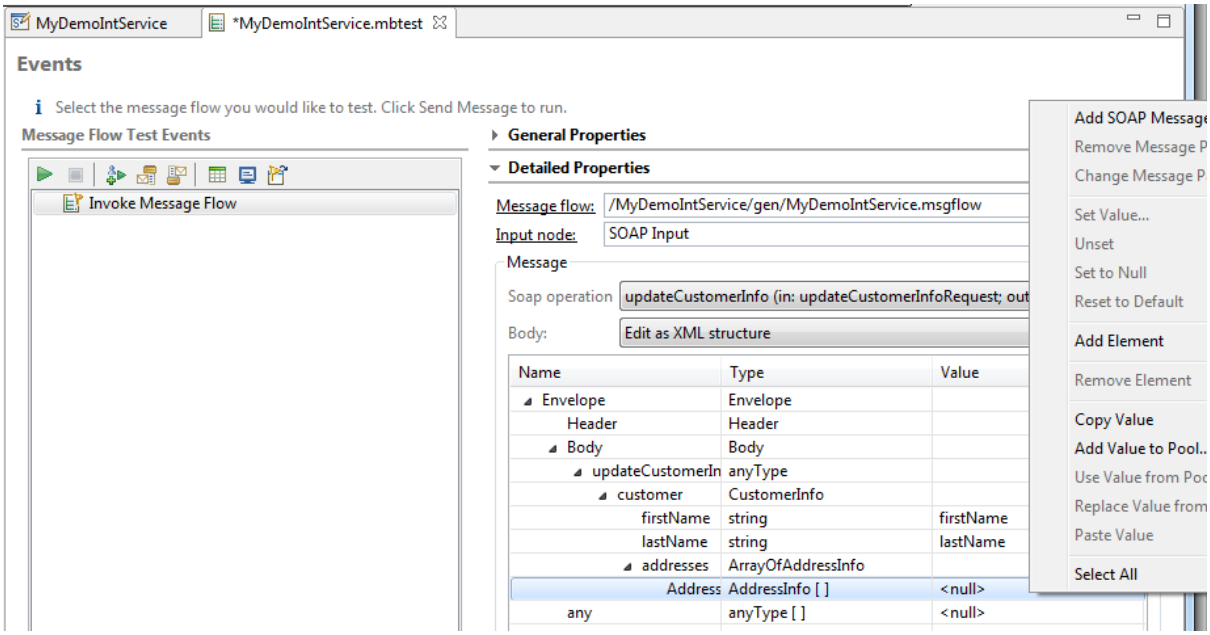


A myDemoIntService.mbtest test client window is opened. The test client populates the test data based on the service definition of the myDemoIntService integration service. Notice it picks up the Transport type, SOAP operation and builds sample input data.

We set some of the input fields with our own data. Right click on the AddressInfo element

AddressInfo	AddressInfo [ ]	<null>

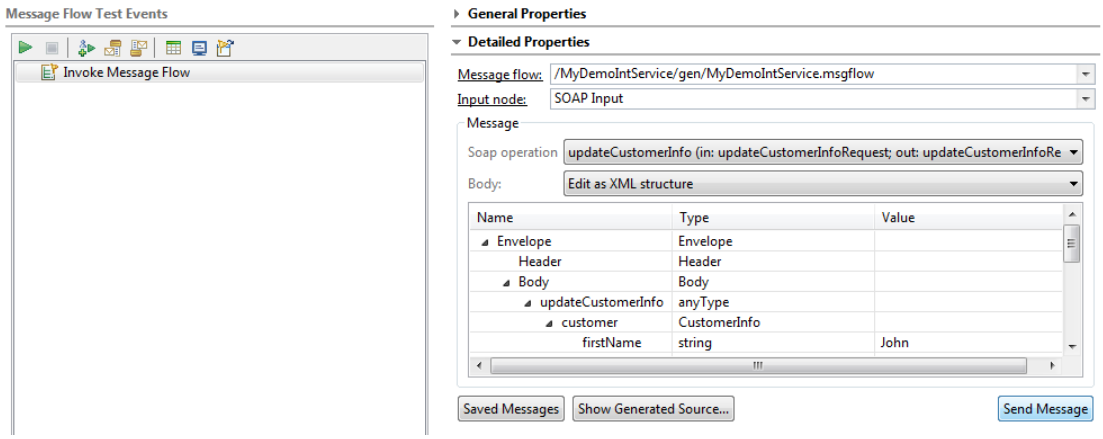
and select Add_Element as shown below.



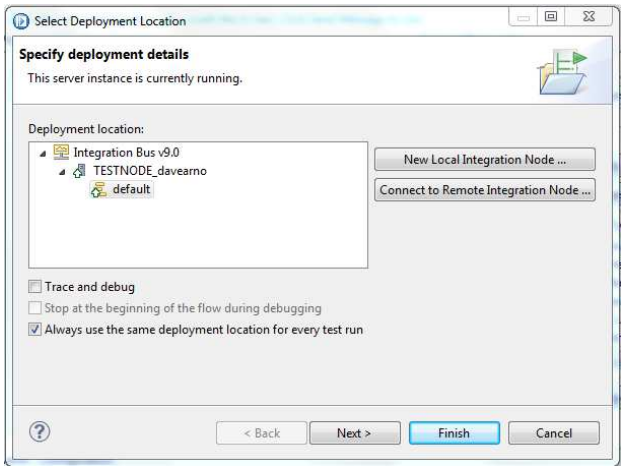
Put in some data for firstName, secondName, zipcode and phone. Leave city and country as-is

Name	Type	Value
Envelope	Envelope	
Header	Header	
Body	Body	
updateCustomerInfo	anyType	
customer	CustomerInfo	
firstName	string	John
lastName	string	Smith
addresses	ArrayOfAddressInfo	
AddressInfo	AddressInfo [ ]	
AddressInfo[0]	AddressInfo	
city	string	city
country	string	country
stateOrProvince	string	stateOrProvince
streetAddress1	string	streetAddress1
zipcode	string	2101
phone	string	12345

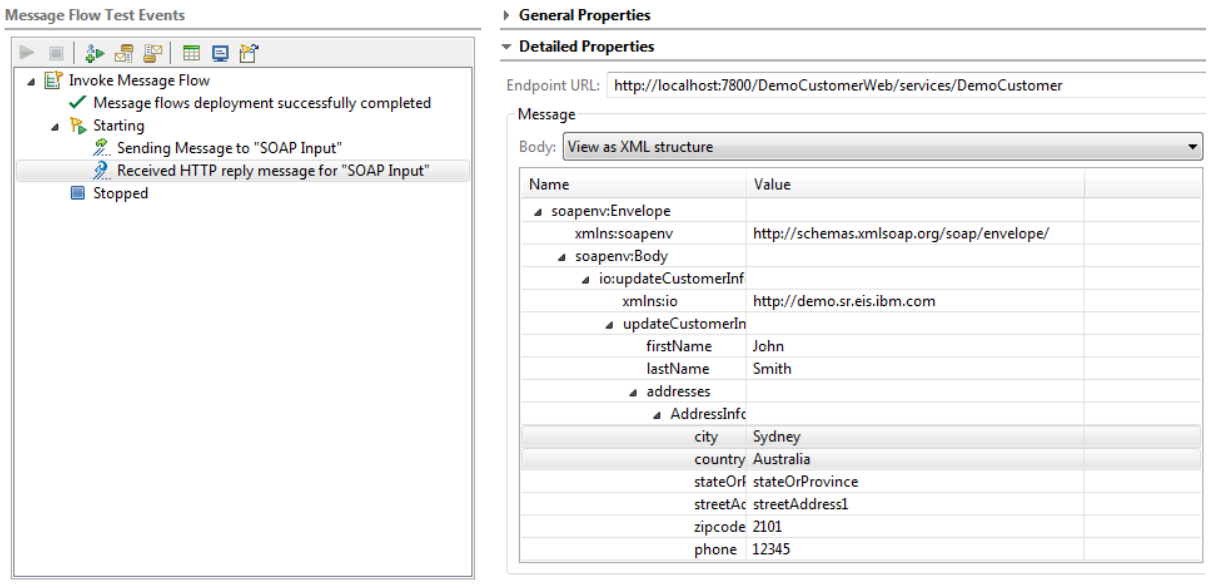
We will now deploy and test. Hit the send message button



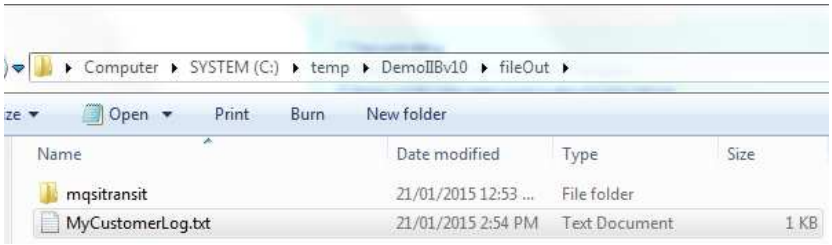
Select the local TESTNODE_*name*->default integration server and click finish

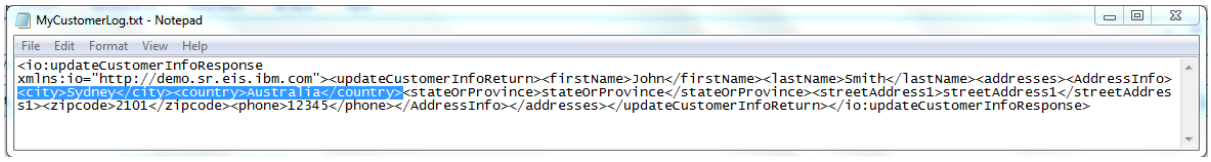


The test client deploys the Integration Service and runs the test. The results should be as shown below.



Finally, let's take a look for the log file in the file system





You have now completed your first build and test of an IBM Integration Bus service