

IBM WebSphere® Business Process Management V7.0 – Lab exercise

WebSphere Integration Developer V7.0

Mapping enhancements in V7

What this exercise is about	2
Lab requirements	2
What you should be able to do	2
Introduction	2
Exercise instructions	3
Part 1: Prepare environment for lab	4
Part 2: Add data map to BPEL flow	7
Part 3: Add If and Else if Conditional Support and use updated XPath 1.0 Expression Builder	13
Part 4: Cut, Copy, and Paste transforms	27
Part 5: Refactor to a submap	33
Part 6: Refactor from a submap	39
Part 7: Data map catalog	41
What you did in this exercise	44
Solution instructions	45

What this exercise is about

The objective of this lab is to highlight V7 XML mapping enhancements and to provide you with an understanding of how to use them. To do this you will take the role of the developer. You have been tasked to add an XML map to a BPEL flow and to map the Address content used by a service with the OrderAddress you are using in your ordering system. The business objects, interface, business process, and assembly diagram have been created for you. Your job will be to build the map using many of the new version 7 XML map features.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Integration Developer V7.0

What you should be able to do

At the end of this lab you should be able to:

- Import a project interchange file into the WebSphere Integration Developer V7.0 development environment
- Create an data map
- Use a data map activity in BPEL
- Use conditional support like if, else if, and else conditions in an XML Map
- Identify the many built-in functions added for XML maps in version 7
- Copy and paste a transform between maps
- Refactor to and from submaps in an XML Map
- Used the data map catalog to find maps faster and for an organized view of maps

Introduction

When you are integrating services, you almost always need to transform the data into a format that the receiving service can process. An XML map defines mappings between input and output business objects or service message objects. You can use the XML map editor to create an XML map between input and output data. An XSL style sheet is generated from the XML map and performs the transformation at runtime. XML maps are also used in with XSL Transformation primitives in mediation flows.

Exercise instructions

Some instructions in this lab are Windows operating system specific. If you plan on running WebSphere Integration Developer on a Linux operating system you will need to run the appropriate commands and use appropriate files for Linux. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference Variable	Windows Location	AIX®/UNIX® Location
<WID_HOME>	C:\IBM\WID70	/opt/IBM/WID70
<WTE_HOME>	C:\IBM\WID7_WTE\runtimes\bi_v7	/opt/IBM/WID7_WTE/runtimes/bi_v7
<WPS_PROFILE_HOME>	<WTE_HOME>\profiles\qmwps	<WTE_HOME>/profiles/qmwps
<LAB_FILES>	C:\Labfiles70\WID\Mapping	/tmp/Labfiles70/WID/Mapping
<WORKSPACE>	<LAB_FILES>\workspace	<LAB_FILES>/workspace
<SOLUTION>	<LAB_FILES>\workspace\solution	<LAB_FILES>/workspace/solution

Windows user note: When directory locations are passed as parameters to a Java™ program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles70\ is replaced by C:/LabFiles70/

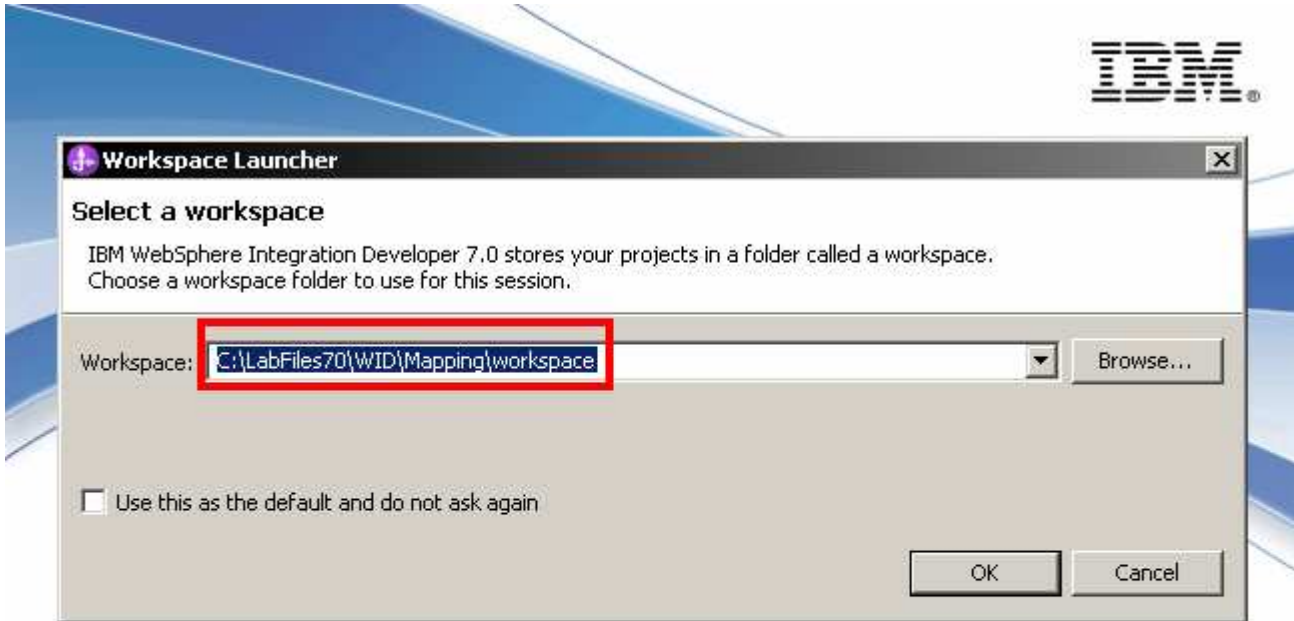
Reference variable	Example: Remote Windows test server location	Example: Remote z/OS® test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	sssr011	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/etc/sscell/AppServer	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<SOAP_PORT>	8880	8880	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	Default	
<USERID>	N/A	Ssadmin	
<PASSWORD>	N/A	fr1day	

Part 1: Prepare environment for lab

To start the lab, you will import a pre-built project in the **WBPMv70_XMLMapLab_PI.zip** project interchange file into a new workspace. The business objects, interface, business process, and assembly diagram have been created for you. Your job will be to build the map using many of the new version 7 XML map features.

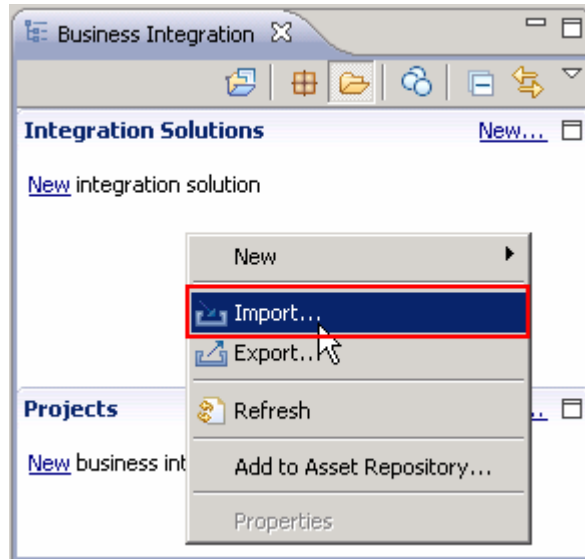
Note: Do not remove the applications from the WebSphere Process Server test environment at this time.

1. Start WebSphere Integration Developer V7.0 with a workspace location of **<WORKSPACE>**

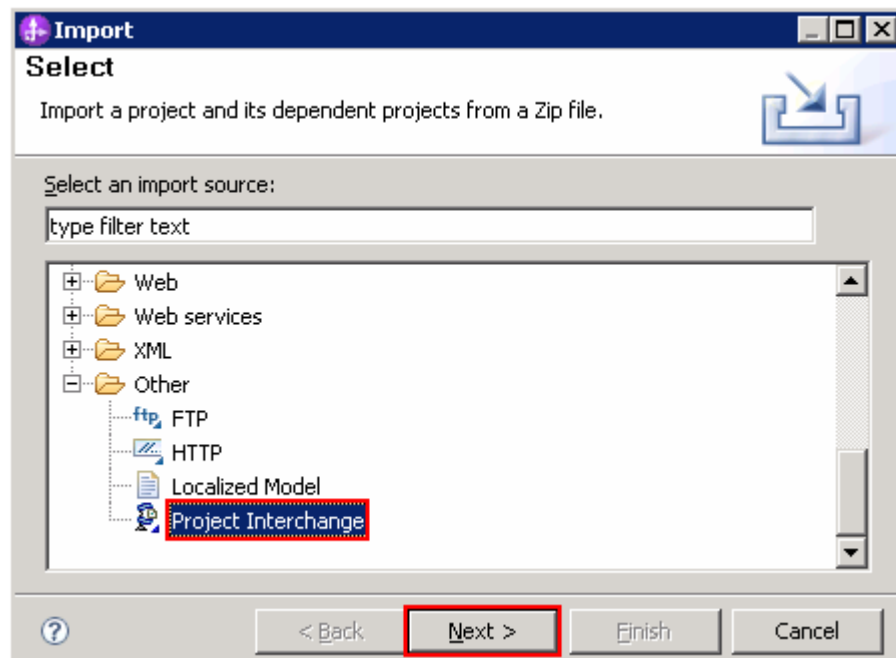


Licensed Materials - Property of IBM Corp. © IBM Corporation and others 1987, 2008.
IBM and the IBM logo are trademarks of IBM Corp. in the United States, other countries, or both.

2. Click **OK**
3. Import Project Interchange file, **WBPMv70_XMLMapLab_PI.zip**, into the development environment
 - a. Right-click inside **Business Integration View** (top left view in the Business Integration Perspective) and select **Import** from the pop-up menu

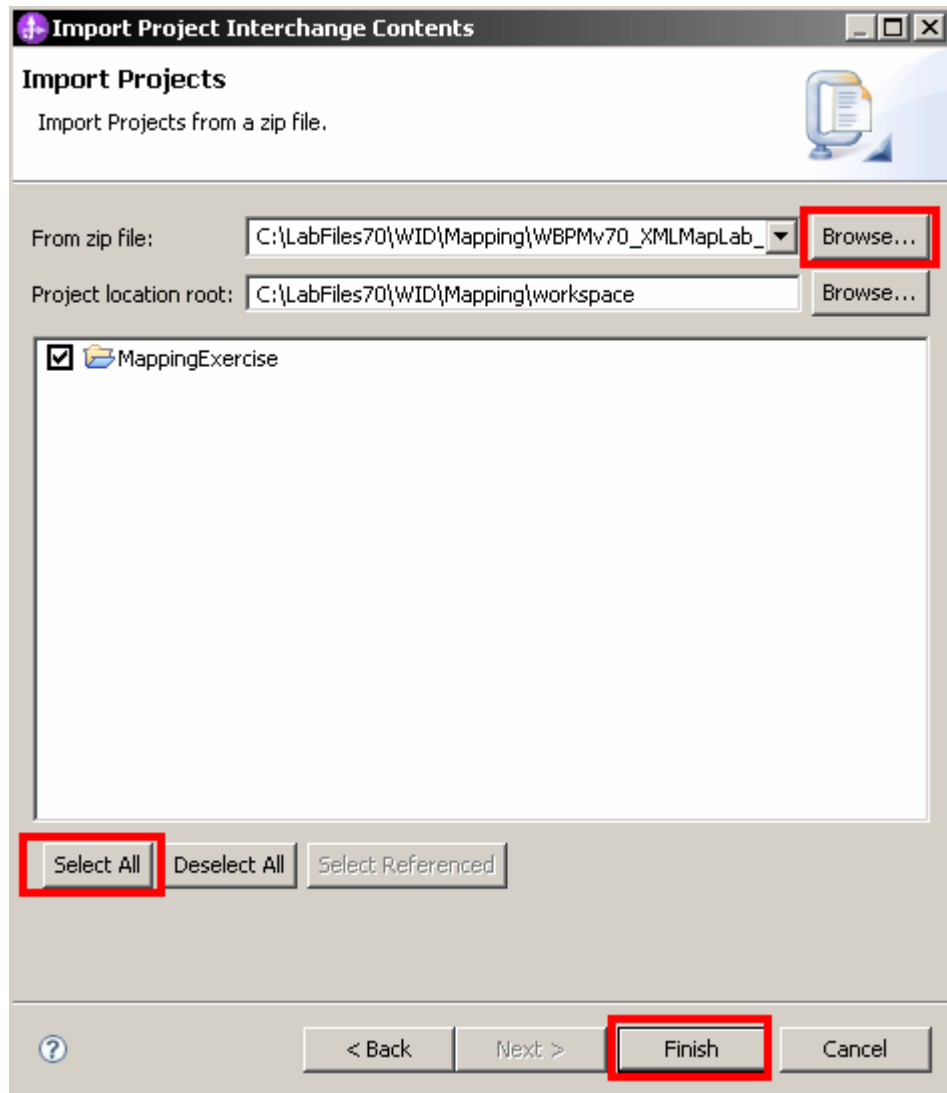


__ b. From the **Import**, dialog, expand **Other** and select **Project Interchange** from the tree



__ c. Click **Next**

__ d. In the next **Import Project** panel, click the **Browse** button for **From zip file** and navigate to **<LAB_FILES>/WBPMv70_XMLMapLab_PI.zip** and click **Open**.

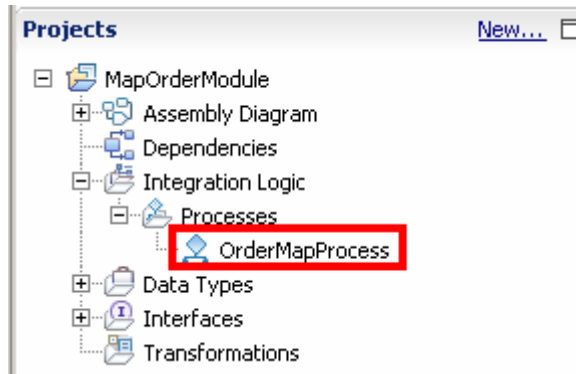


- ___ e. Click the **Select All** button to select all the check boxes for the projects listed
- ___ f. Click **Finish** button (projects are imported and auto-build will run).
- ___ g. Verify you have **MappingExercise** module listed in the Business Integration view under the **Projects** section.

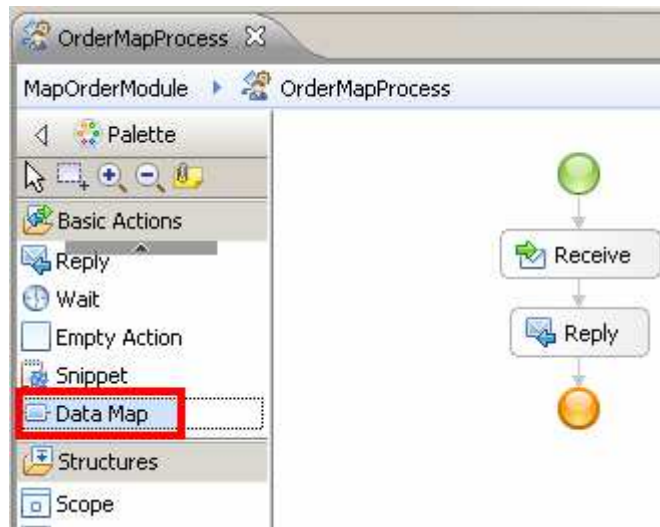
Part 2: Add data map to BPEL flow

In this part of the lab, you will add an XML map to a BPEL flow. The new data map wizard converges the business object map and XML map wizards into one wizard to help you decide which type of map you want to use. Text and additional help will guide you to making an informed choice. In addition, the XML Map now available in BPEL as a data map activity. Previously, the business object map was only map available to be used in BPEL. You will also go through some XML Map toolbar options. The toolbar options are not new for V7, but a useful reminder will be valuable.

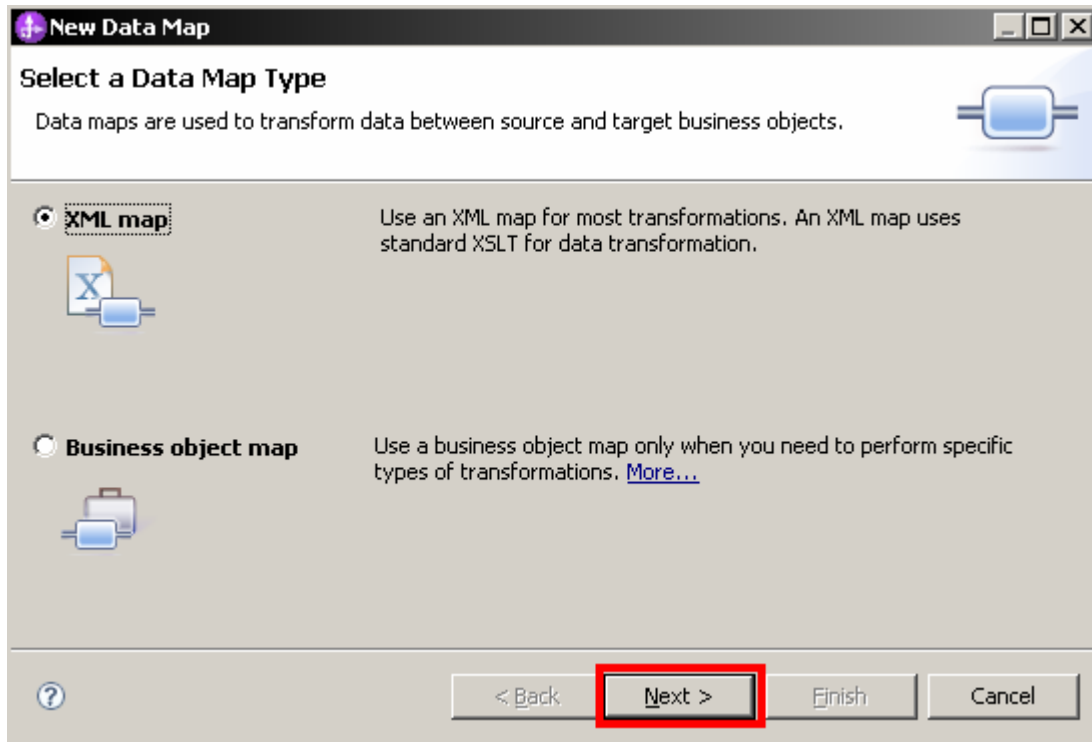
- ___ 1. Open the OrderMapProcess and add a data map.
 - ___ a. In the Business Integration view, click the + icon next to Integration Logic and drill down to **Integration Logic > Processes > OrderMapProcess** and double click the **OrderMapProcess**.



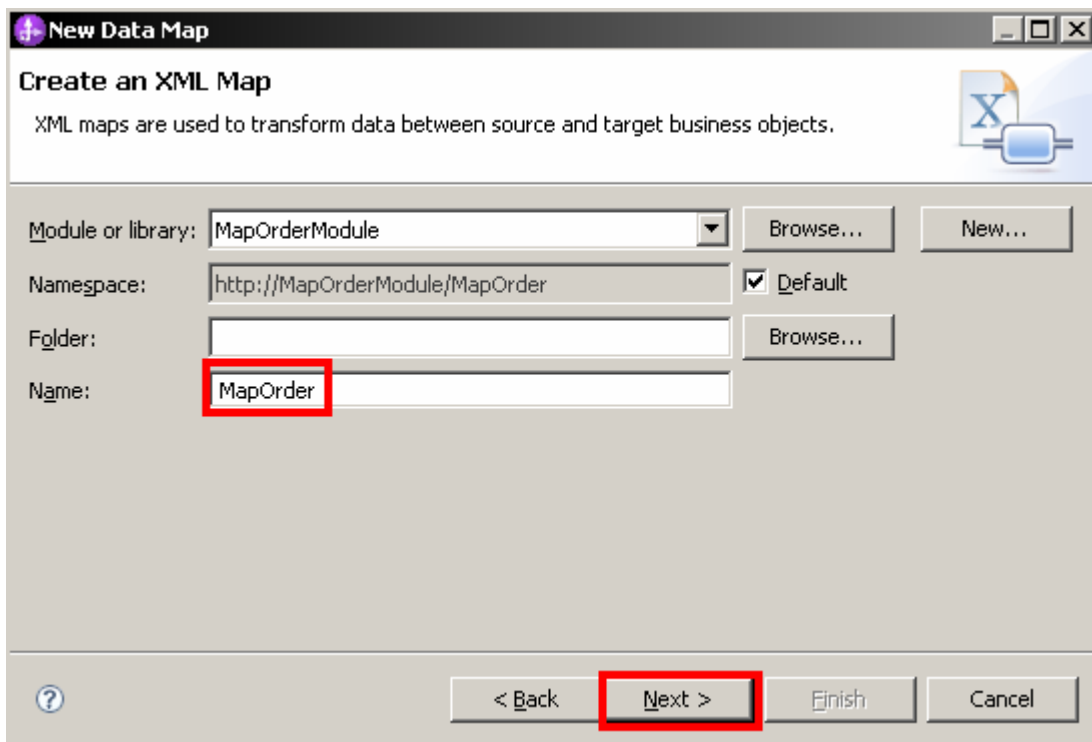
- ___ b. Once open, you will notice only a receive and reply activity. Select the **Data Map** activity in the Palette.



- ___ c. Drag and drop the Data Map activity between the Receive and Reply activities in the BPEL flow. You will be presented with the **New Data Map** wizard.

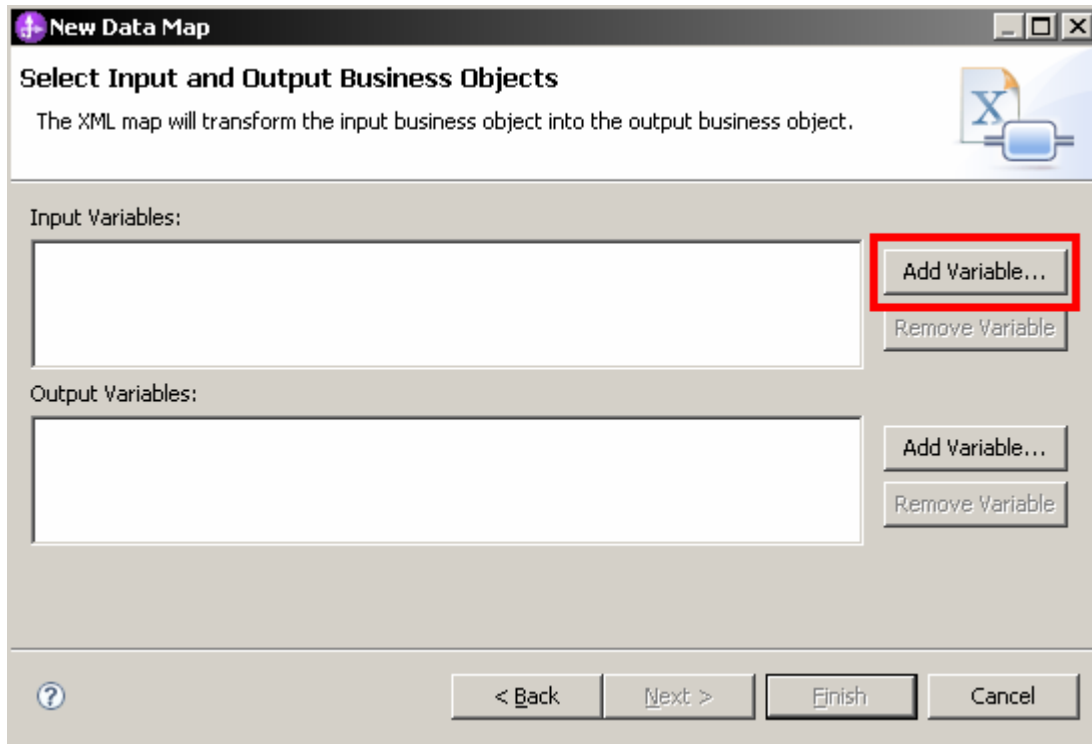


- ___ d. Click **Next**.
- ___ e. Type **MapOrder** in for the name of the map.

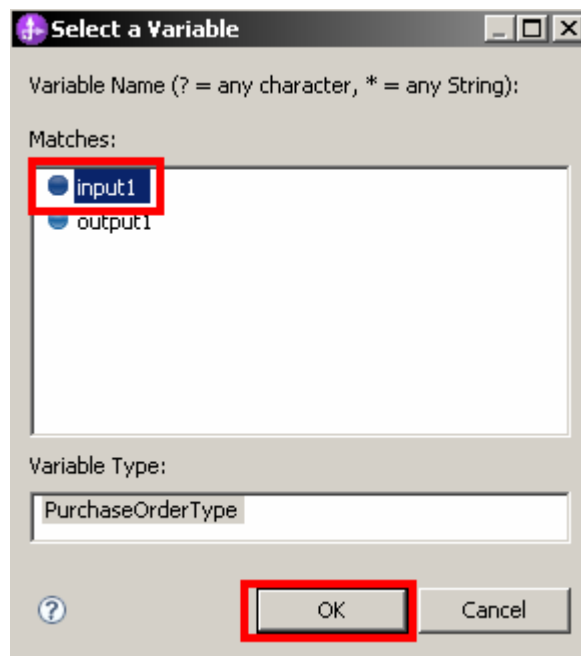


___ f. Click **Next**.

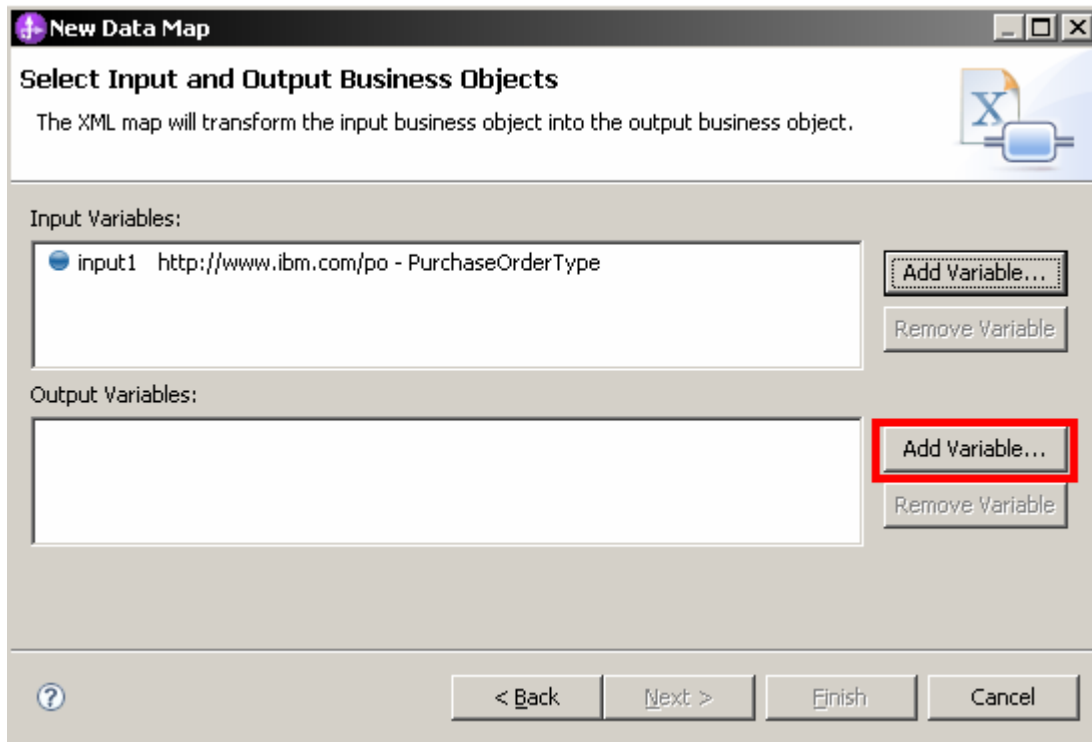
___ g. Click the **Add Variable** button next to the **Input Variables** area



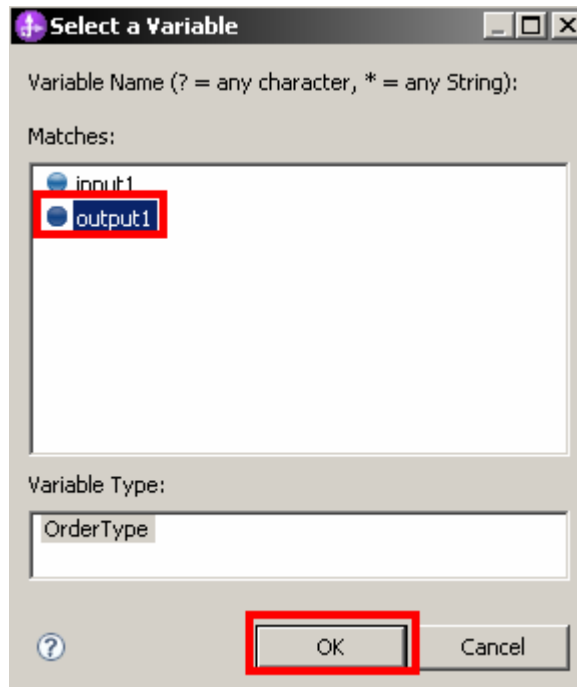
___ h. Select the **input1** variable and click **OK**.



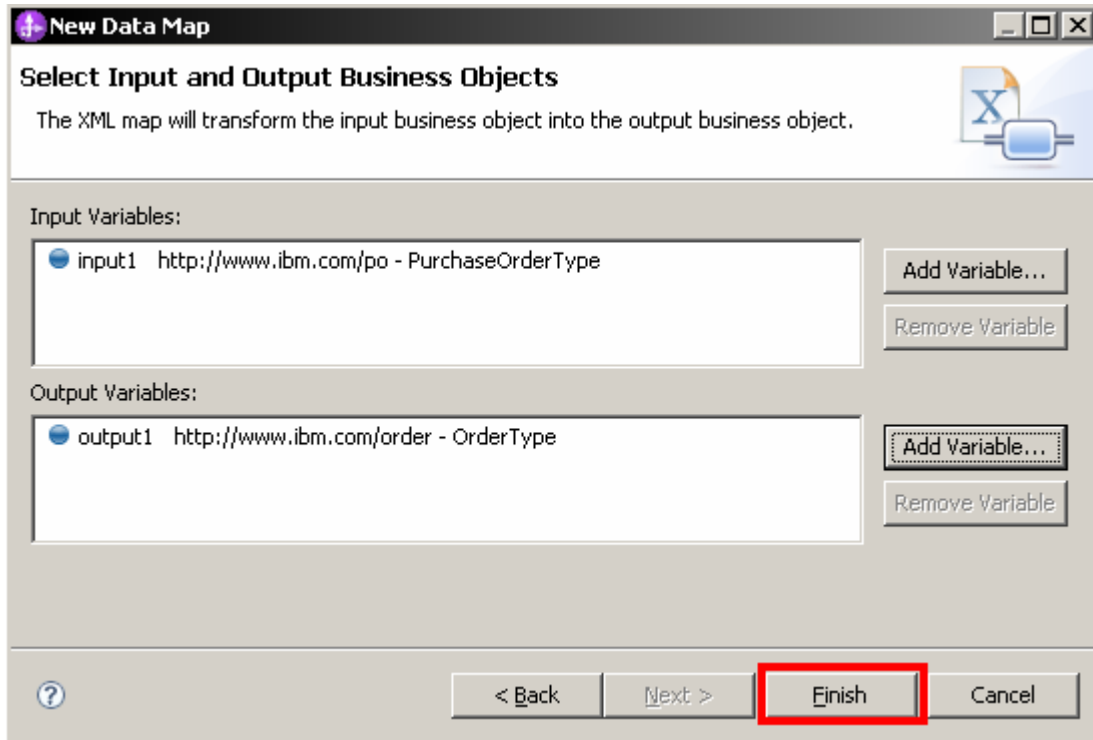
___ i. Click the **Add Variable** button next to the **Output Variables** area



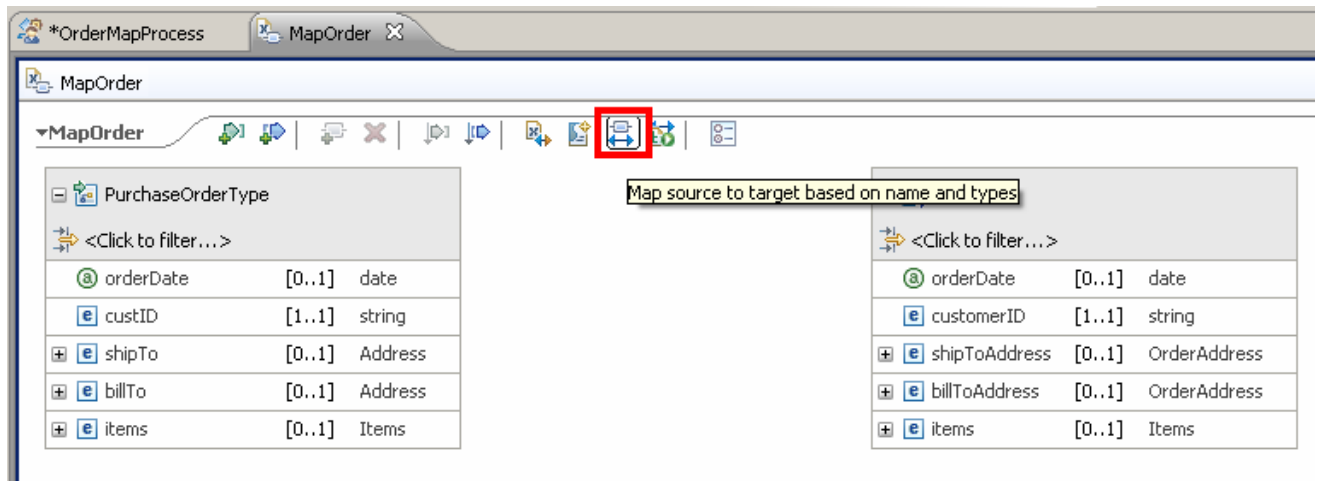
___ j. Select the **output1** variable and click **OK**.



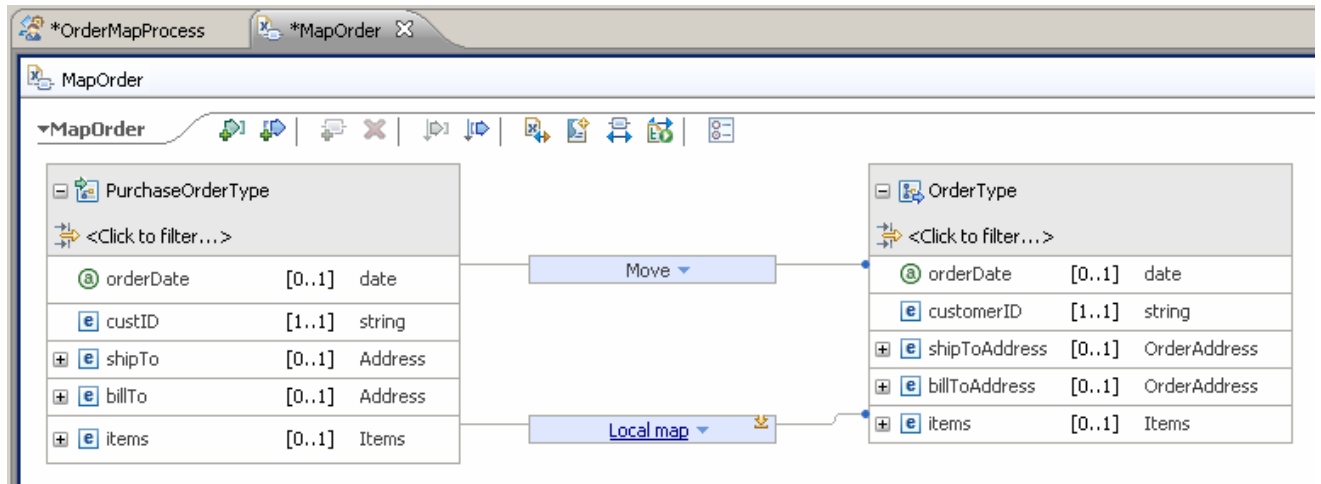
___ k. Click **Finish** once you see the input and output types have been assigned.



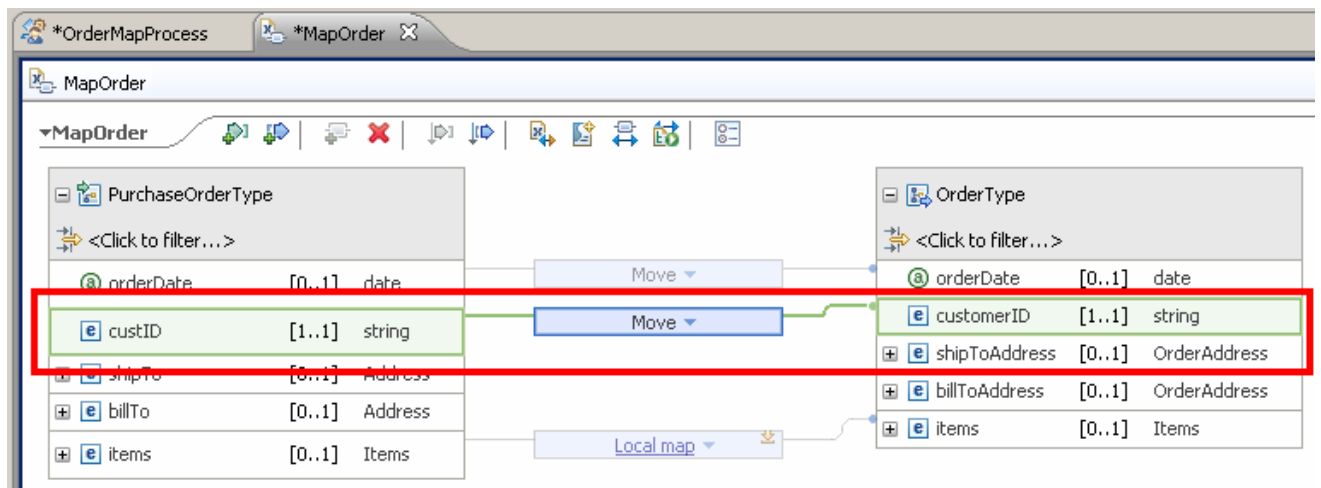
___ l. Once finished, your map will be built with the source object (left hand side) and target object (right hand side). Notice some of the elements are of the same name and types. Use the toolbar option to automatically create transforms for those that are same name and type. **Click the Map source to target based on name and types icon.**



- ___ m. A move transform will automatically will be generated for the orderdate elements and a LocalMap transform will automatically will be generated for the array of items.



- ___ n. Add a move transform to connect the custID and customerID elements. Select the **custID** element on the source business artifact (left side) and drag a wire to **customerID** on the target business artifact (right side).

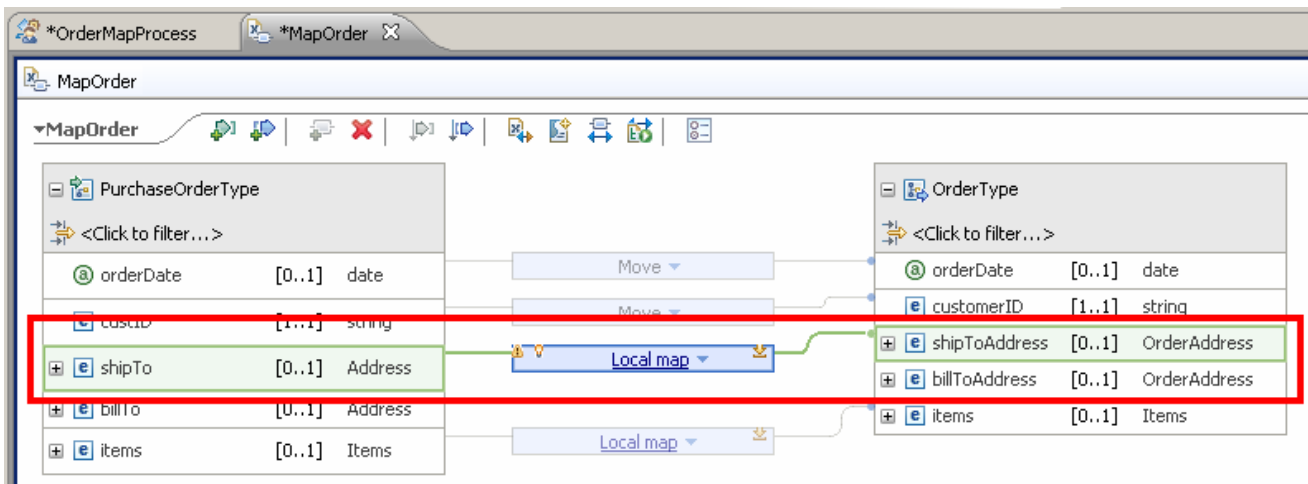


- ___ o. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

Part 3: Add If and Else if Conditional Support and use updated XPath 1.0 Expression Builder

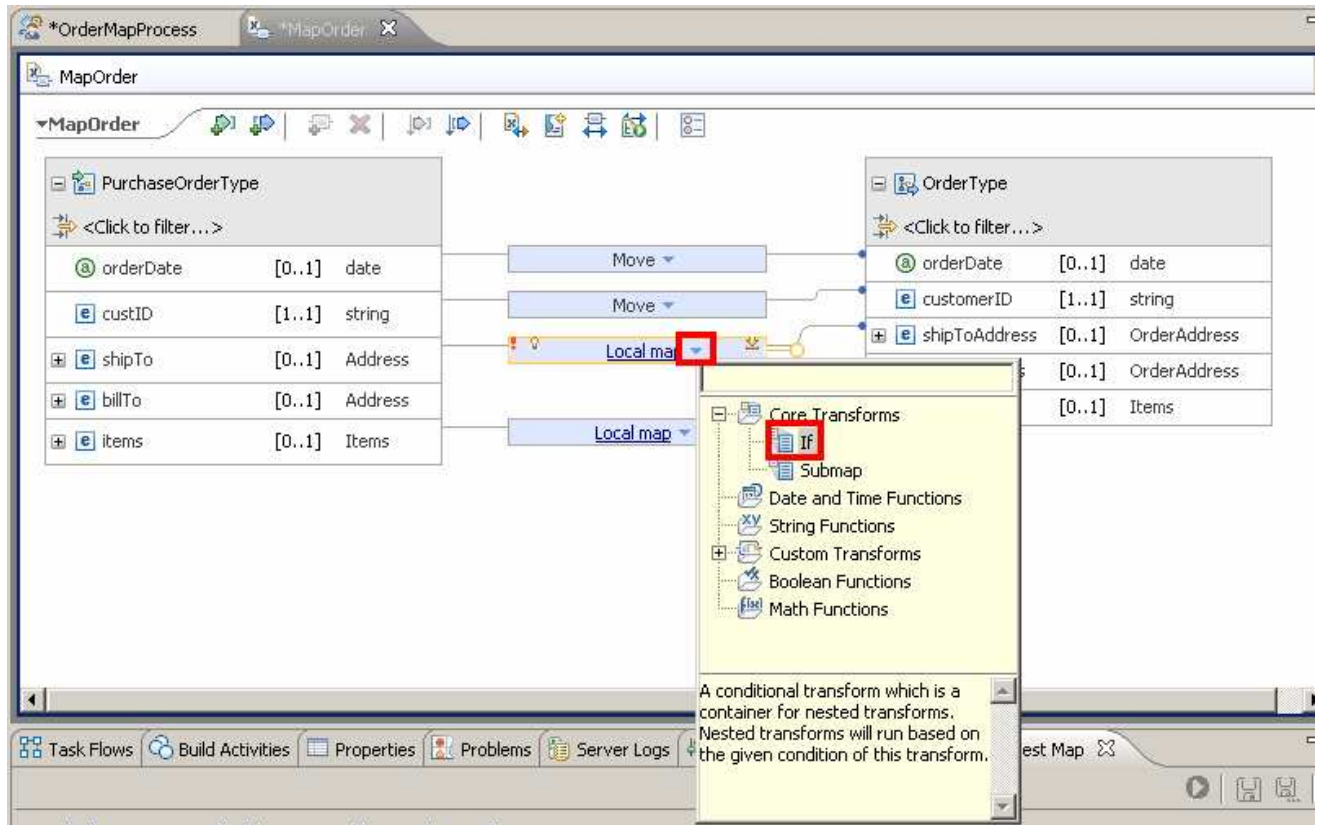
In this part of the lab, you will explore the conditional support and the XPath 1.0 expression builder improved in version 7. Conditional support provides the ability to make logic decisions in an XML Map. WebSphere Integration Developer version 6.2.0.1 only supported the “if” condition. Now in version 7 the if, else if, and else conditions are supported. Logic is evaluated in order from top to bottom and fires the transform when the condition in the transform is true. To build the condition, the XPath 1.0 expression builder has been improved in version 7. The initial text area can be used to copy/paste any previously built XPath 1.0 expressions into the text area. Content assist and validation are automatically on in the initial text area, but you can also click the button to open the simple Xpath expression builder. This means more complex Xpath expressions can be created easier. The simple XPath expression builder provides the ability to drill down or filter to find elements of the business object you are working with. You can also add optional filters or conditions on those elements and will receive immediate validation on the expression while you build it. In addition, you have easier access to the XPath 1.0 Expression Builder from the BPEL Editor, Mediation Flow Editor, and XML map editor.

- ___ 1. Add a local map transform to connect the shipTo and shipToAddress elements and open the local map.
 - ___ a. Select the **shipTo** element on the source business artifact (left side) and drag a wire to **shipToAddress** on the target business artifact (right-side). A local map will be created automatically.

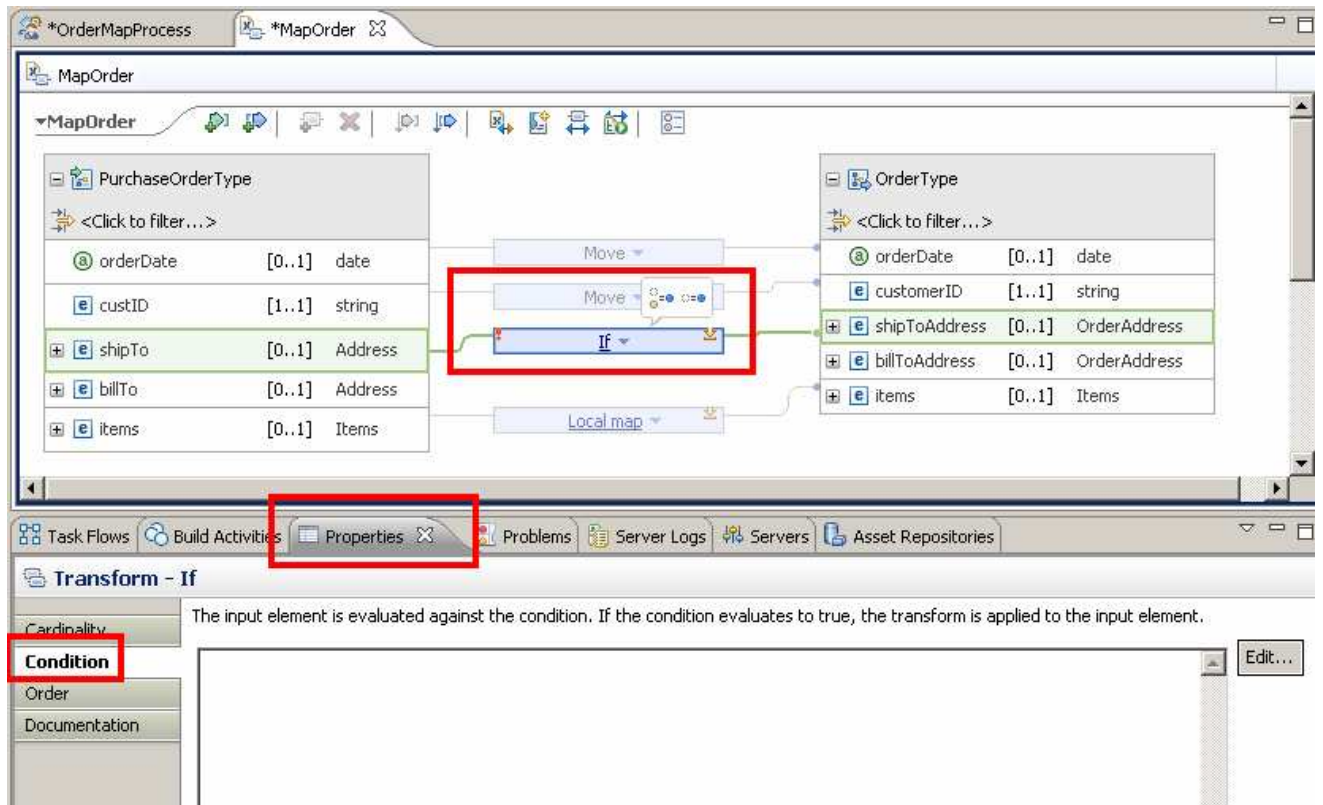


- ___ 2. Add an If condition transform. Change the Local Map transform to an If condition transform.
 - ___ a. Click the **small blue triangle** to the right of Local map in the transform to bring up the built-in function menu. The menu is dynamic, meaning it will only show you applicable transforms for the elements you are mapping.
 - ___ b. Type “if” in the filter text area at the top of the built-in function menu to only show built-in functions with that text. Only one transform appears, the If. Notice the description area at the bottom of the built-in function menu providing you instant information on what the transform does.

- ___ c. **Click the If option** in the built-in functions menu to change the Local Map transform to an If condition transform.

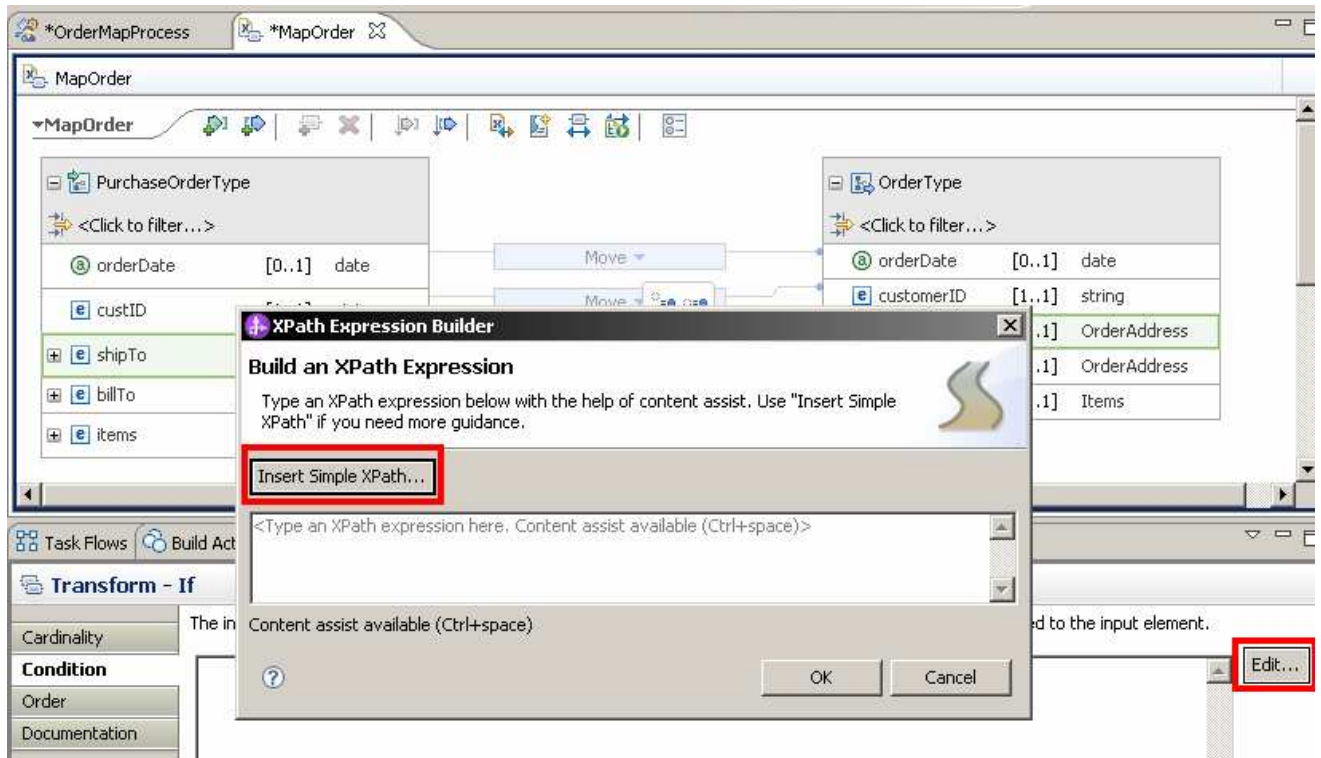


- ___ 3. Add XPath 1.0 expression to provide the condition logic.
 - ___ a. Select the **If transform** so that it is highlighted/has focus.
 - ___ b. Click the **Properties view**
 - ___ c. Click the **Condition tab** inside of the Properties view.



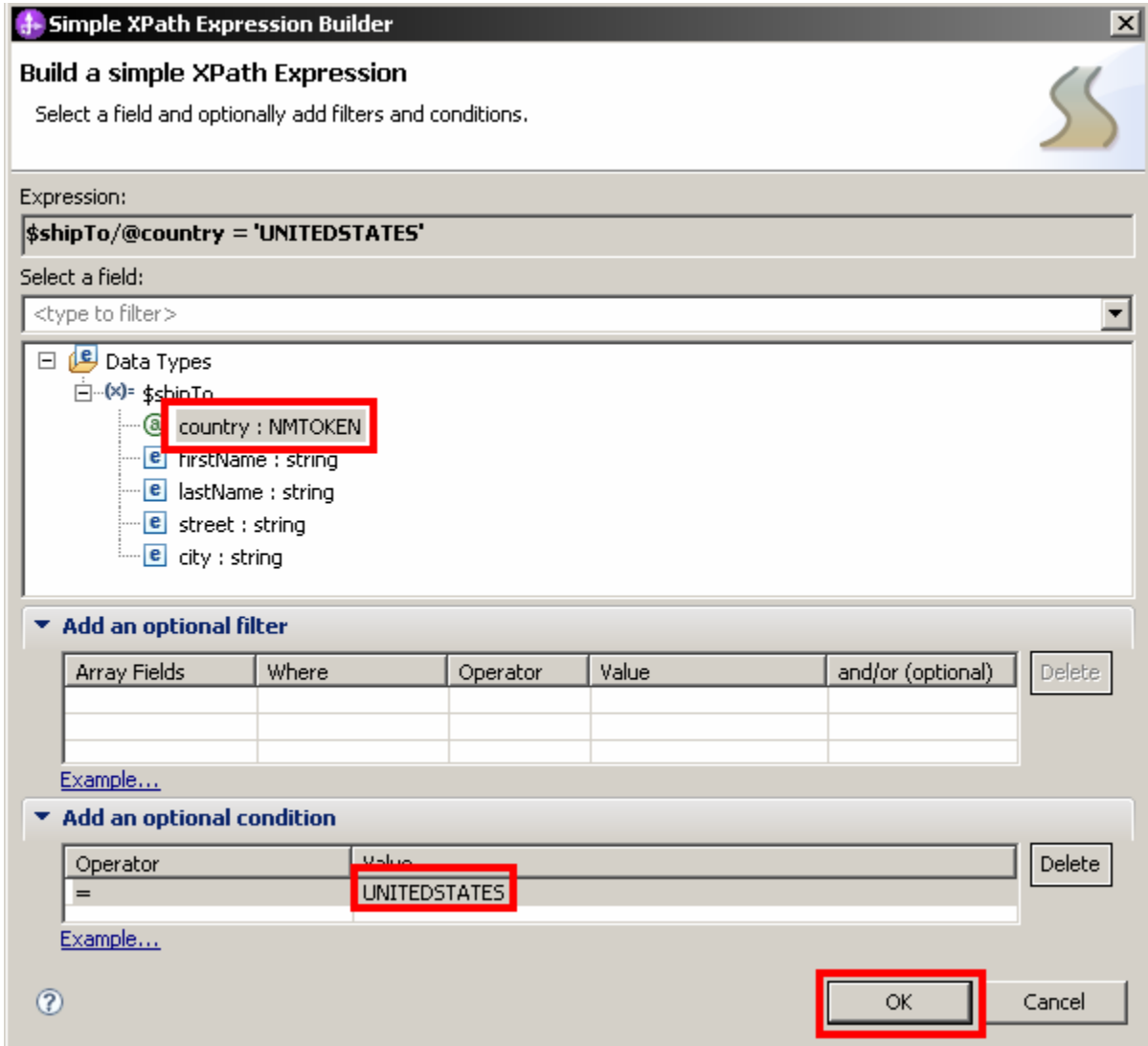
___ d. Click the **Edit** button on the right side to create an expression condition.

- e. You could start typing in your condition here in the text area and have instant content assist and validation, but click the **Insert Simple XPath** button to open the XPath 1.0 Expression Builder.



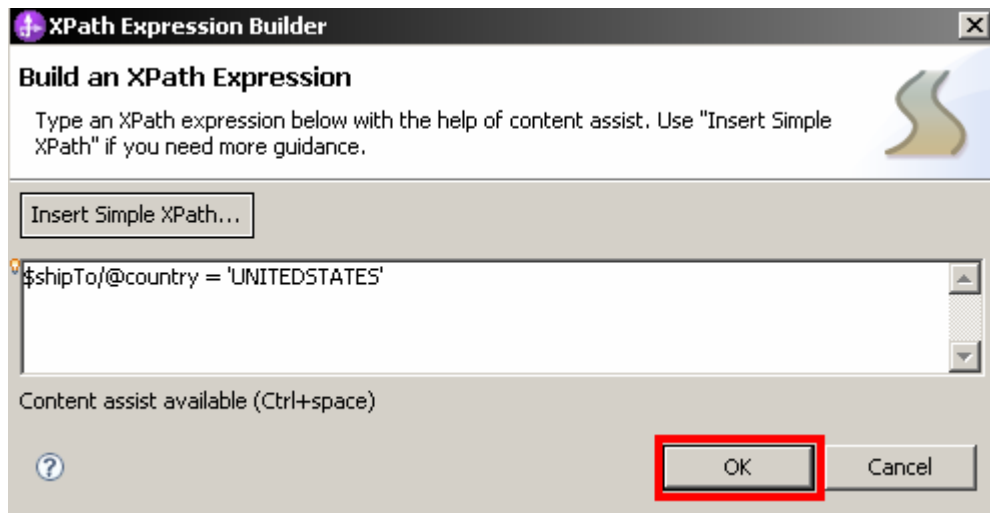
- f. In the “select a field:” area, either start typing “country” in the <type to filter> text box or drill down in the business artifact by clicking the + symbol next to \$shipTo. **Click “country” element** to select and add it to the expression.

- g. Under “Add an optional filter” area, leave the Operator as “=”, but **add the text United States** to the **Value** text box.



- h. Notice that the expression is being built at the top of the builder (in bold text). Also, you can add an optional filter for array fields. **Click OK.**

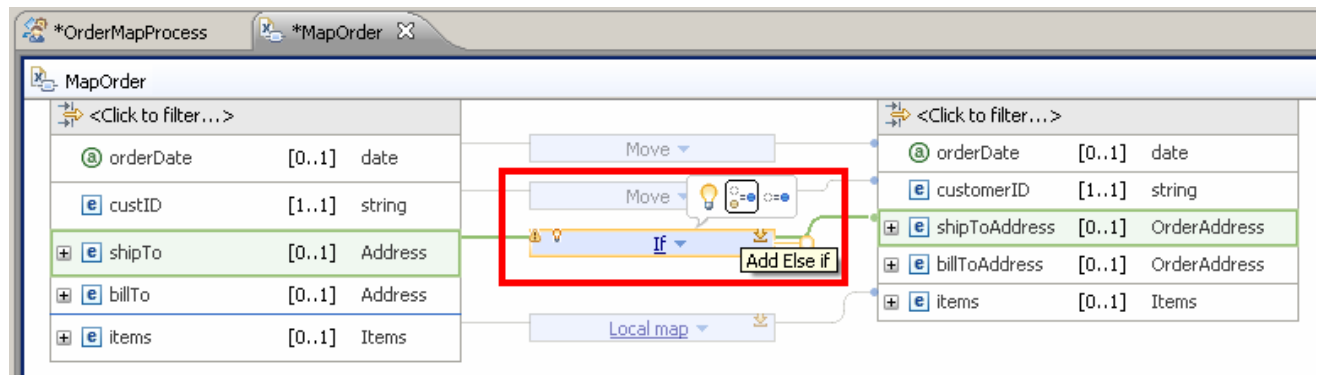
- ___ i. Once back in the main Expression Builder, notice that you can now make any manual changes. This is the expression that you want, so **click OK**.



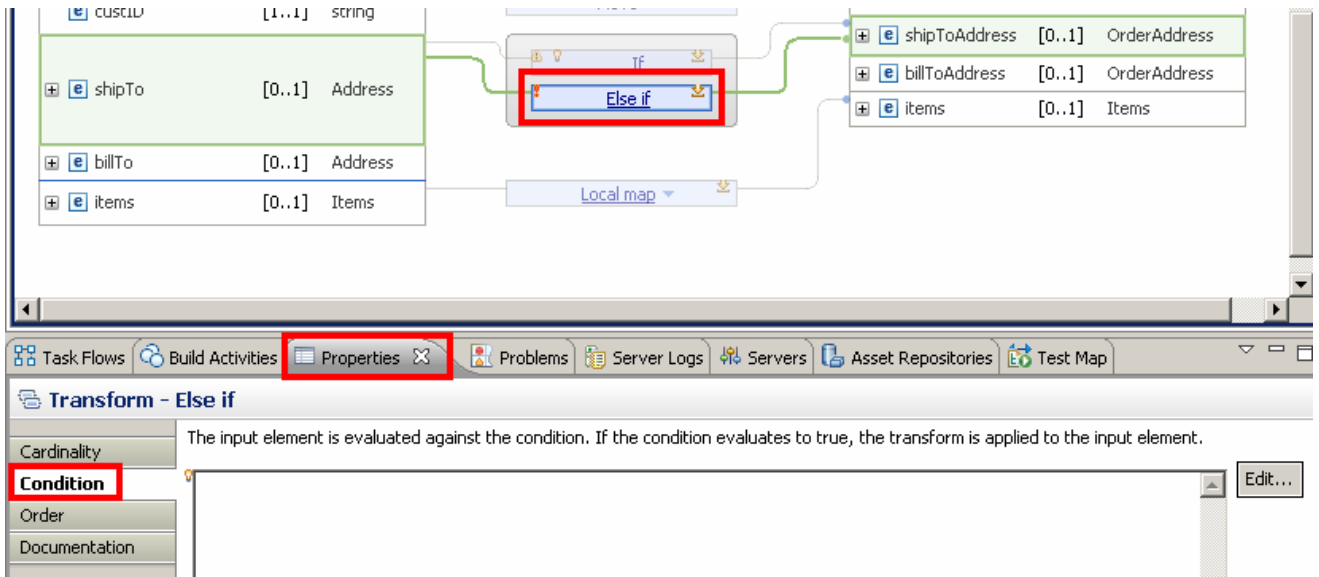
- ___ j. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

___ 4. Add an Else-If condition

- ___ a. Reselect If transform, click "Add Else if" icon that appears on the hover over menu

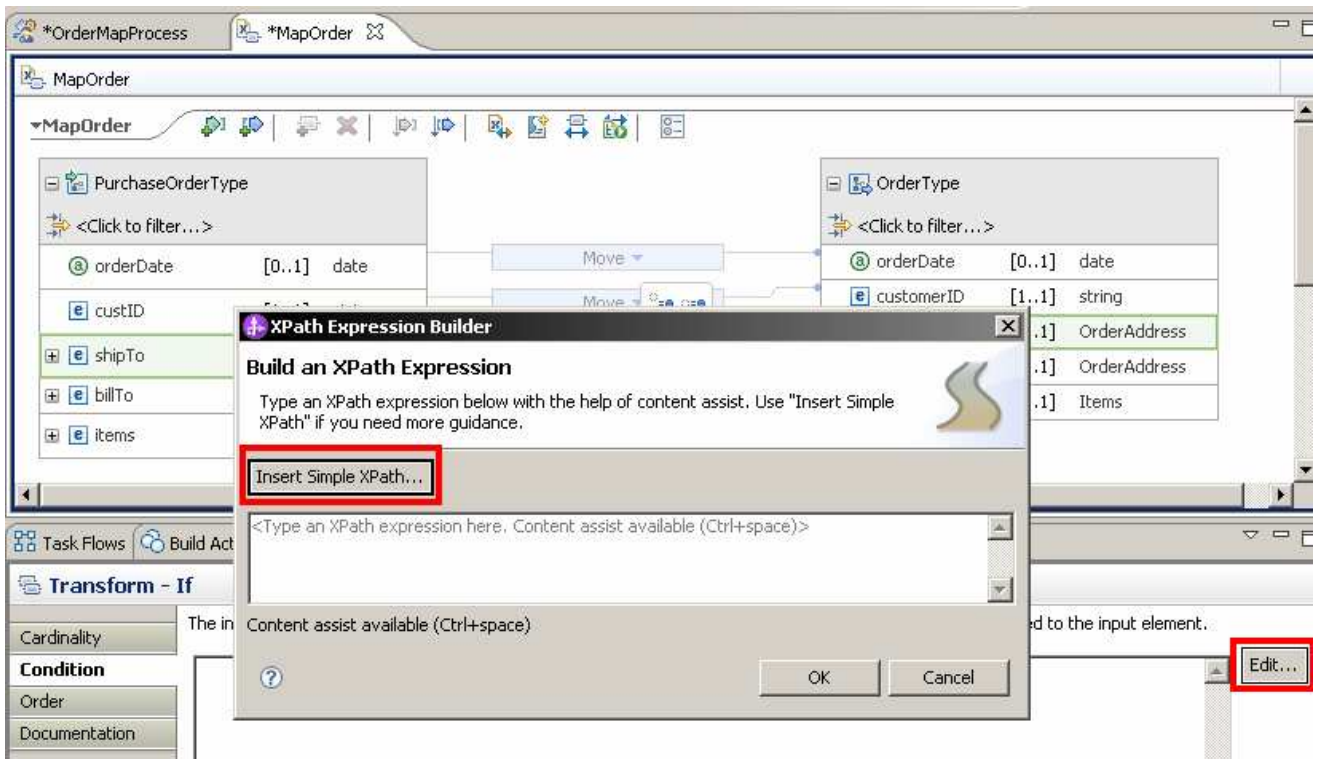


b. Select Else if transform and open Properties View → Condition

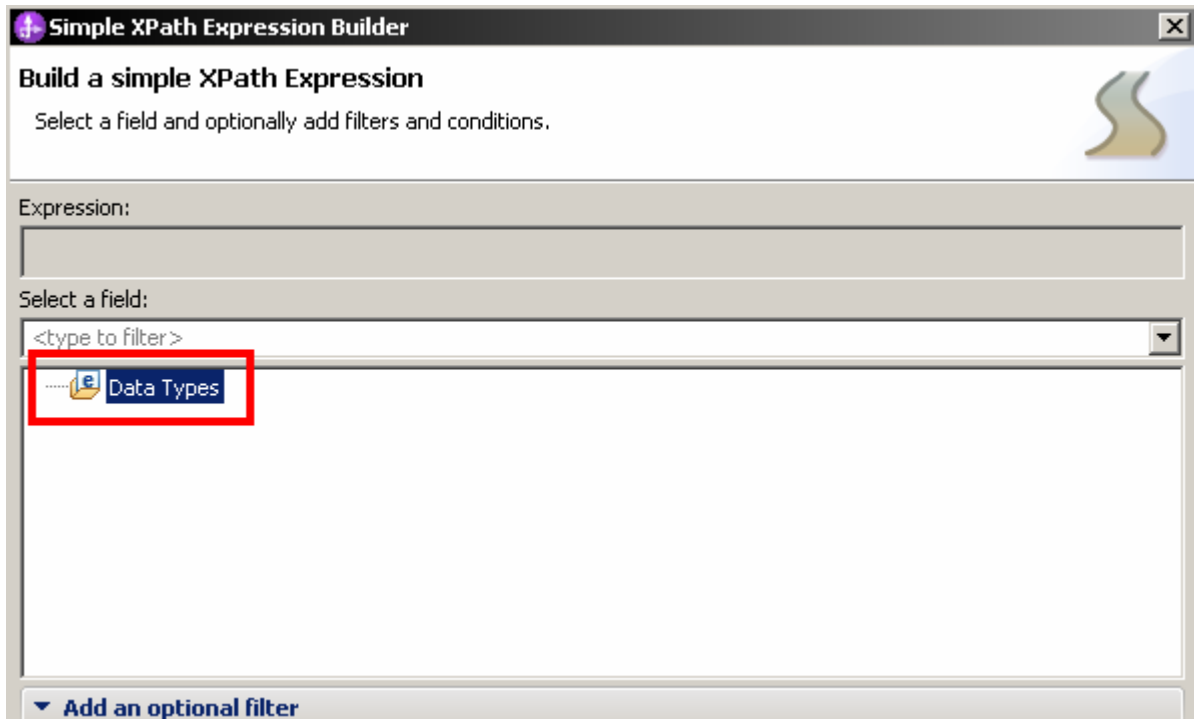


c. Click the Edit button on the right side to create an expression condition.

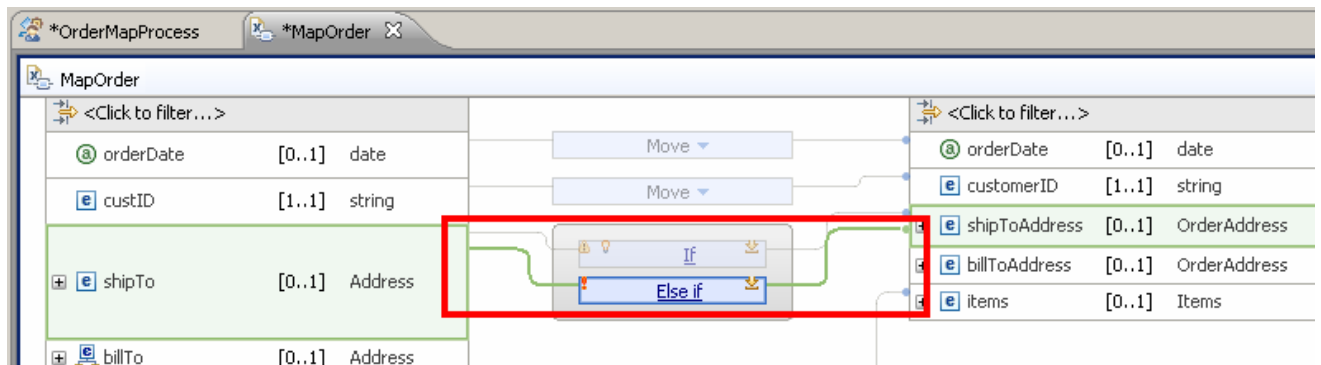
d. Click the Insert Simple XPath button to open the XPath 1.0 Expression Builder.



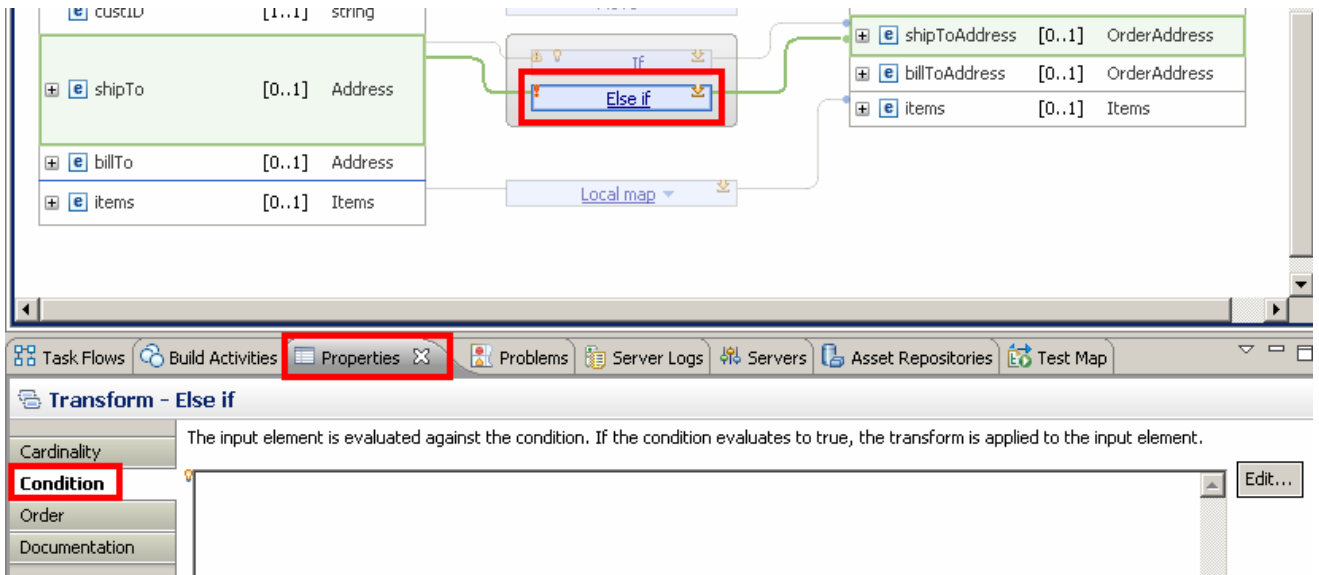
- ___ e. Notice there is no data type to drill down on. This is because a wire is not connecting the Else-If transform from source to target. The reason the Else-If is not automatically connected is because the transform could point to a different source or target element.



- ___ f. Next action is to go back and wire the Else-If transform. Click **Cancel** and **Cancel** again.
- ___ g. In the map, Select the **shipTo** element on the source business artifact (left side) and drag a wire to the Else-If transform. Then click the Else-If transform on the right side and connect the wire to **shipToAddress** on the target business artifact (right side).

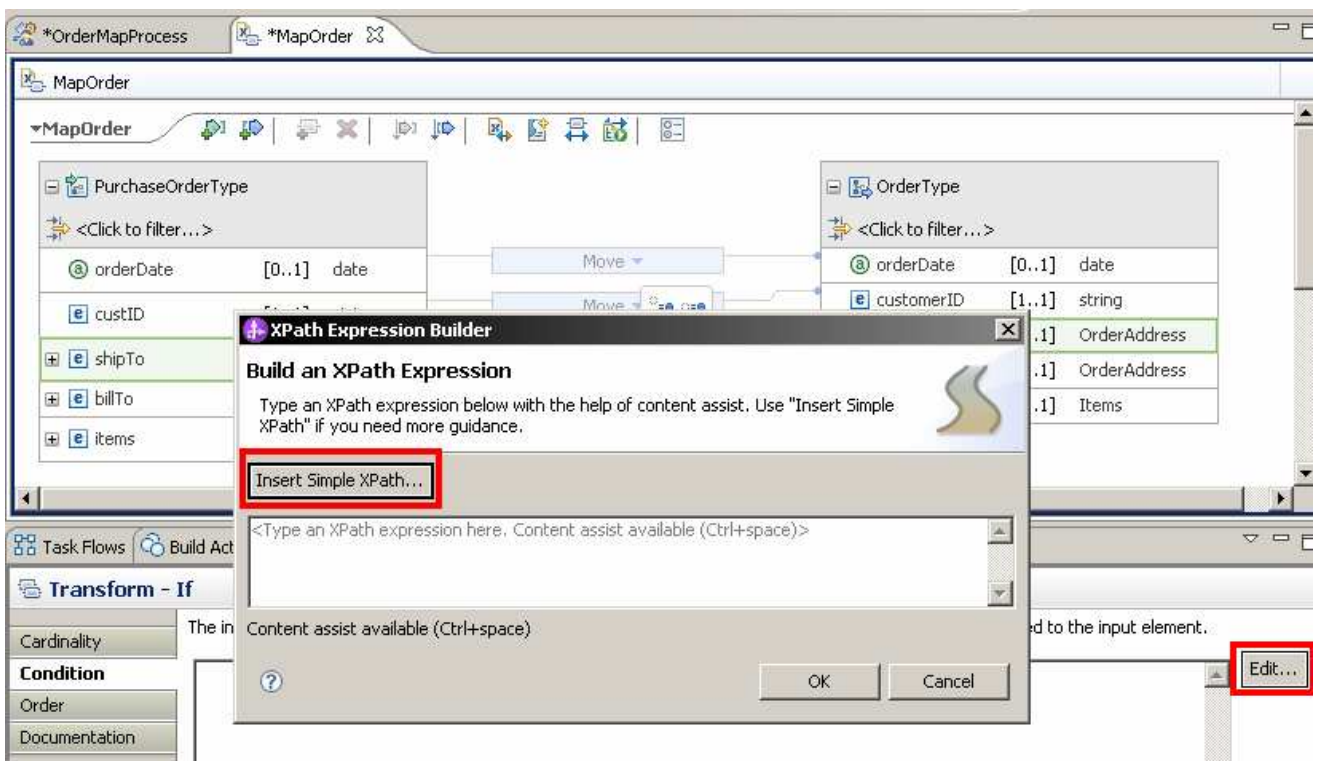


- ___ 5. Add XPath 1.0 expression to provide the condition logic.
 - ___ a. **Select the Else-If transform** so that it is highlight/has focus.
 - ___ b. Click the **Properties** view
 - ___ c. Click the **Condition** tab inside of the Properties view.



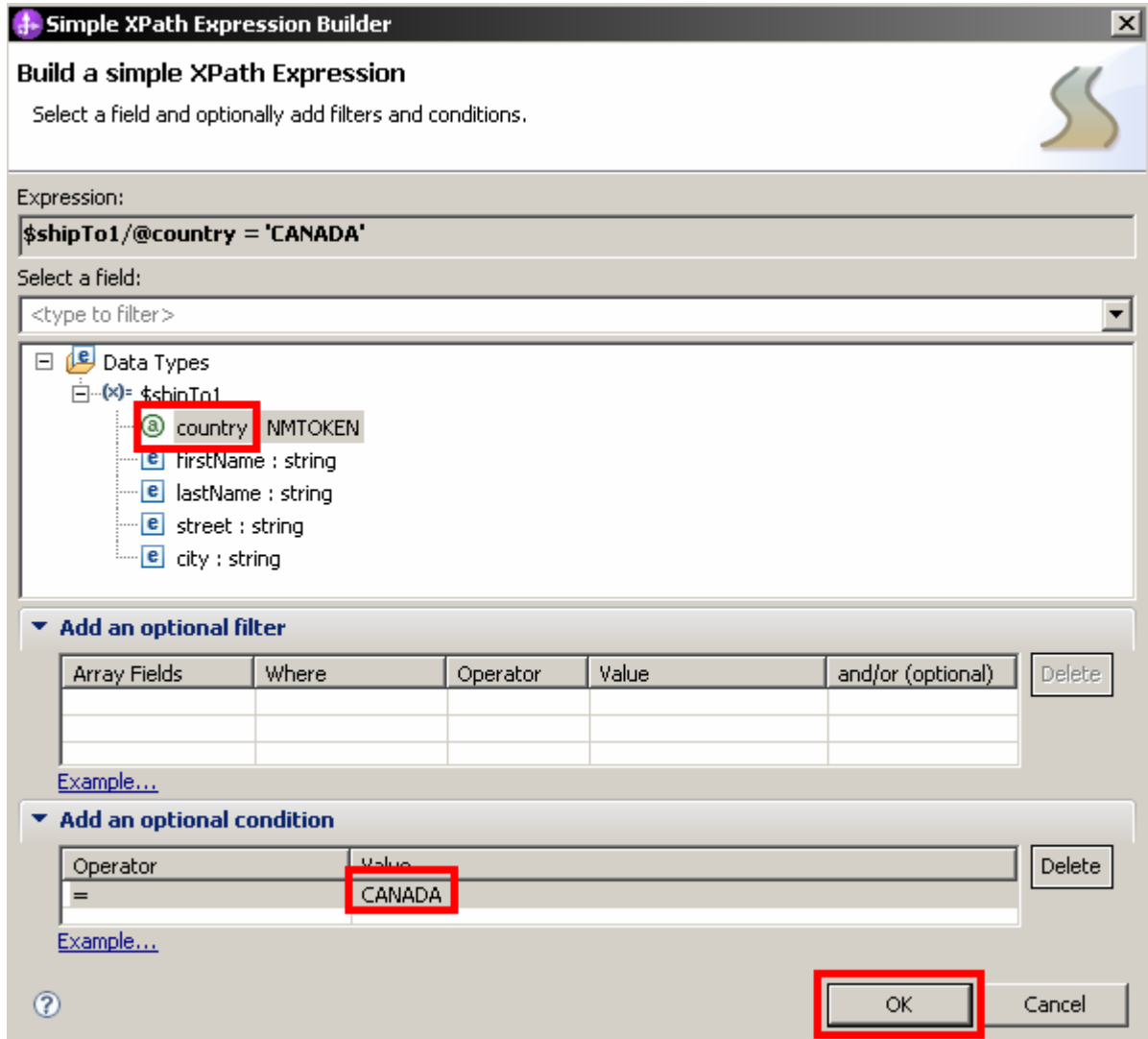
___ d. Click the **Edit button** on the right side to create an expression condition.

___ e. You could start typing in your condition here in the text area and have instant content assist and validation, but click the **Insert Simple XPath button** to open the XPath 1.0 Expression Builder.



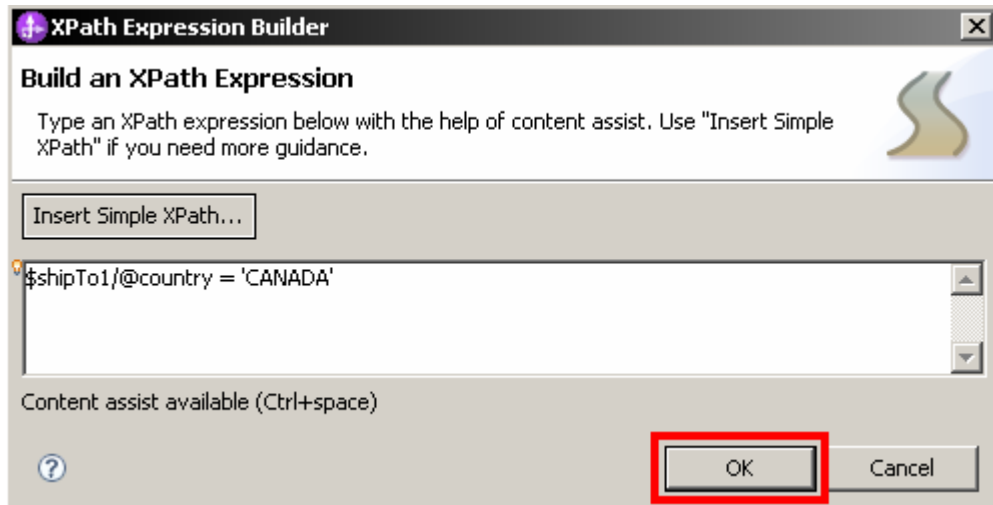
___ f. In the “select a field:” area, either start typing “country” in the <type to filter> text box or drill down in the business artifact by clicking the + symbol next to \$shipTo. Click **“country” element** to select and add it to the expression.

- ___ g. Under “Add an optional filter” area, leave the Operator as “=”, but **add the text CANADA** to the **Value** text box.



- ___ h. Notice that the expression is being built at the top of the builder (in bold text). **Click OK.**

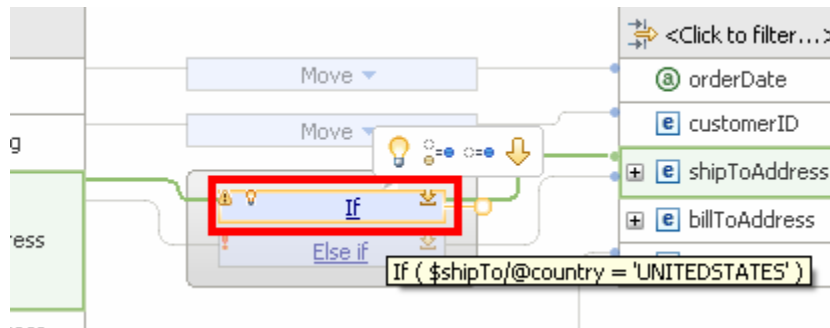
- ___ i. Once back in the main Expression Builder, notice that you can now make any manual changes. This is the expression that you want, so **click OK**.



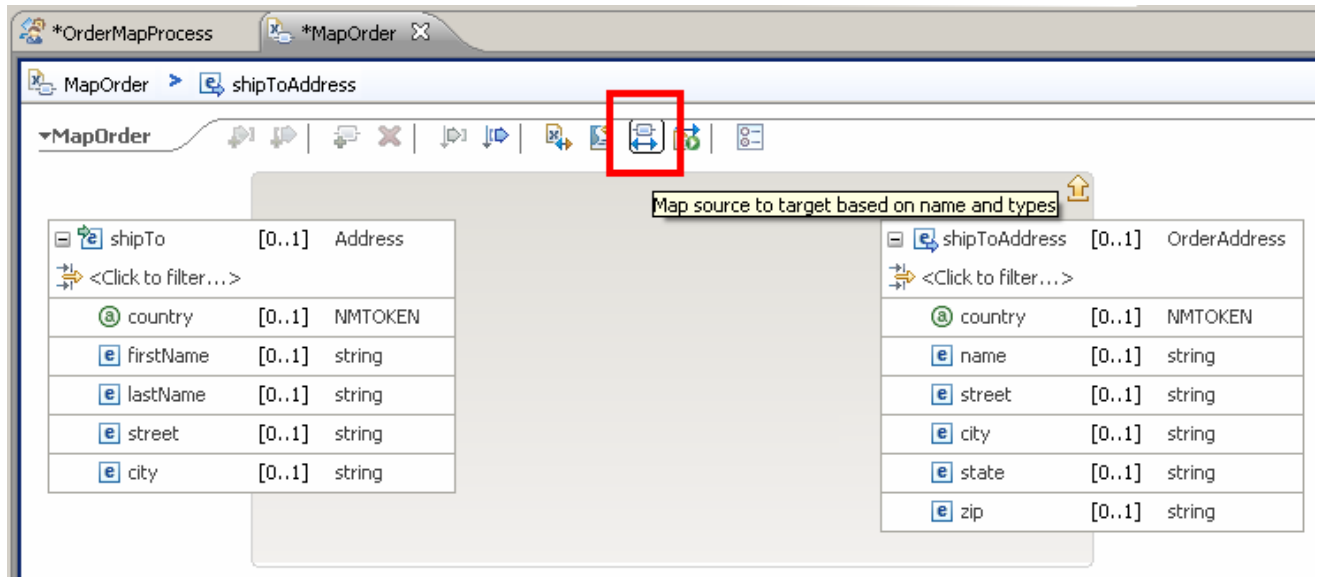
- ___ j. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

___ 6. Finish map inside of If transform

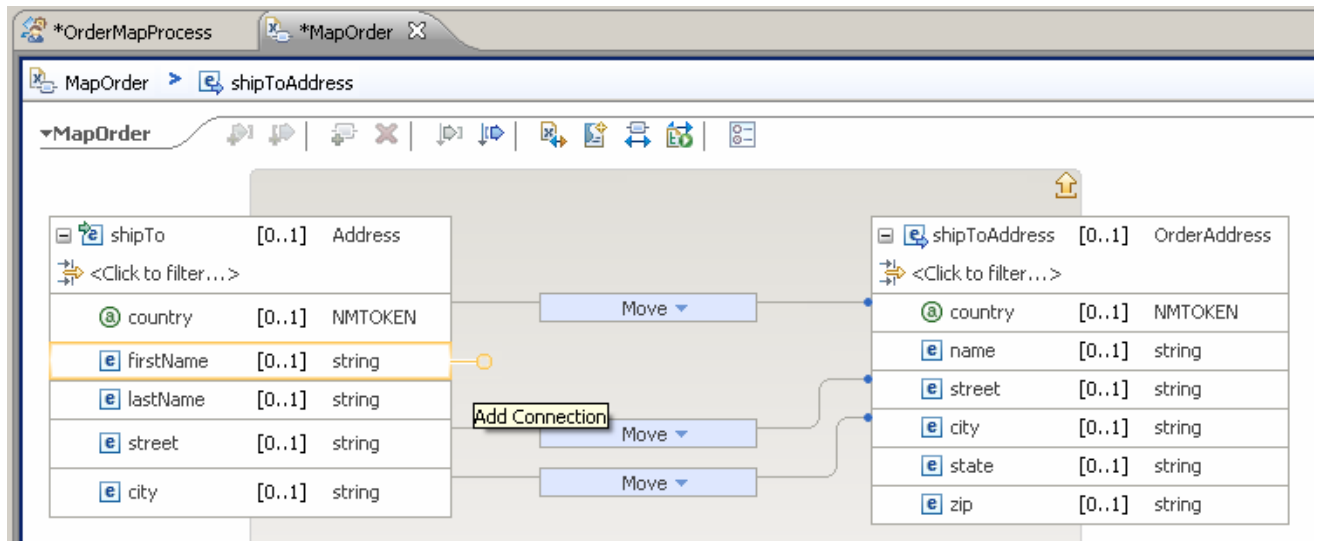
- ___ a. Click the **If text** (which is a link) or **double click the If transform** to open the map



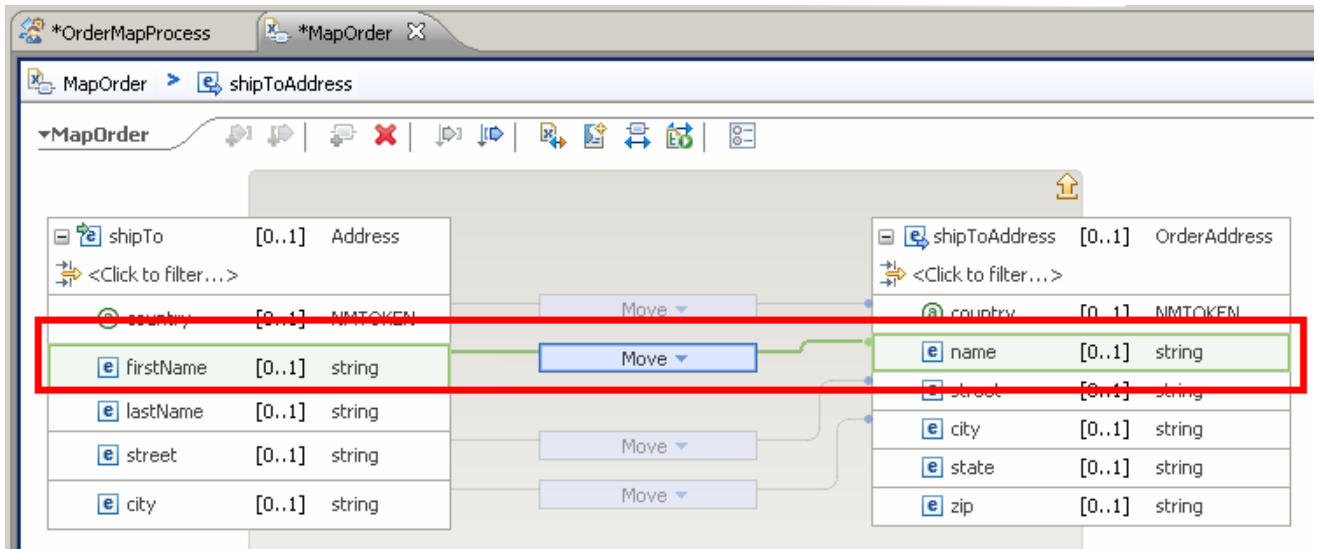
- ___ b. Given the source and target elements are named the same, click the **Map source to target based on name and types button** to automatically create transforms.



- ___ c. . Country, street, and city will be automatically mapped with a Move transform. The firstName, lastName, and name fields will not be automatically mapped since the names do not match. State and zip elements do not have a matching element in the source.

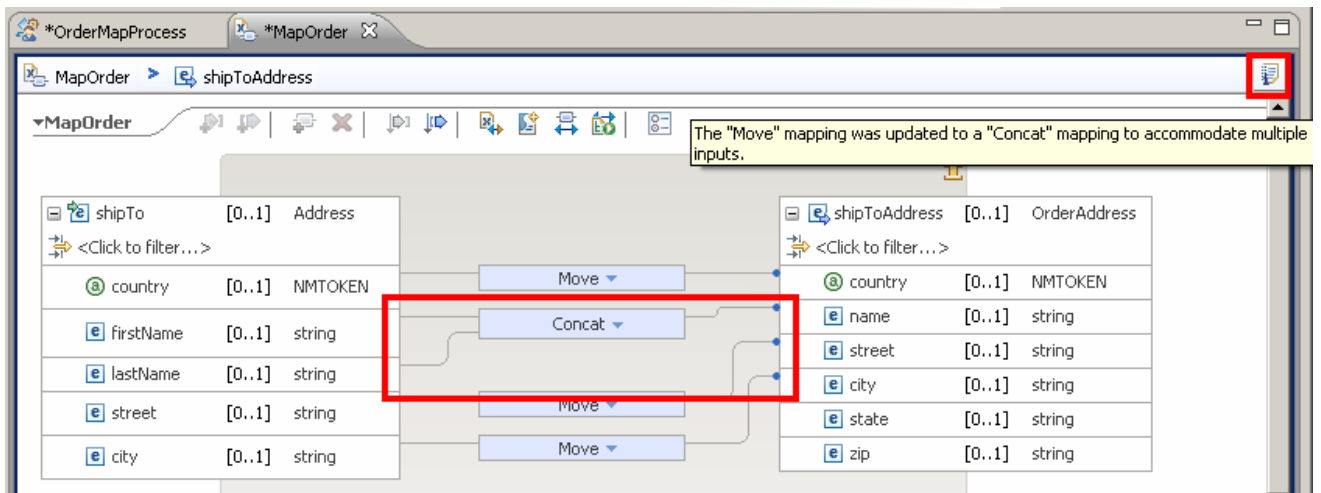


- d. To finish the map manually, select **FirstName** in the source artifact and drag to **name** in the target artifact.

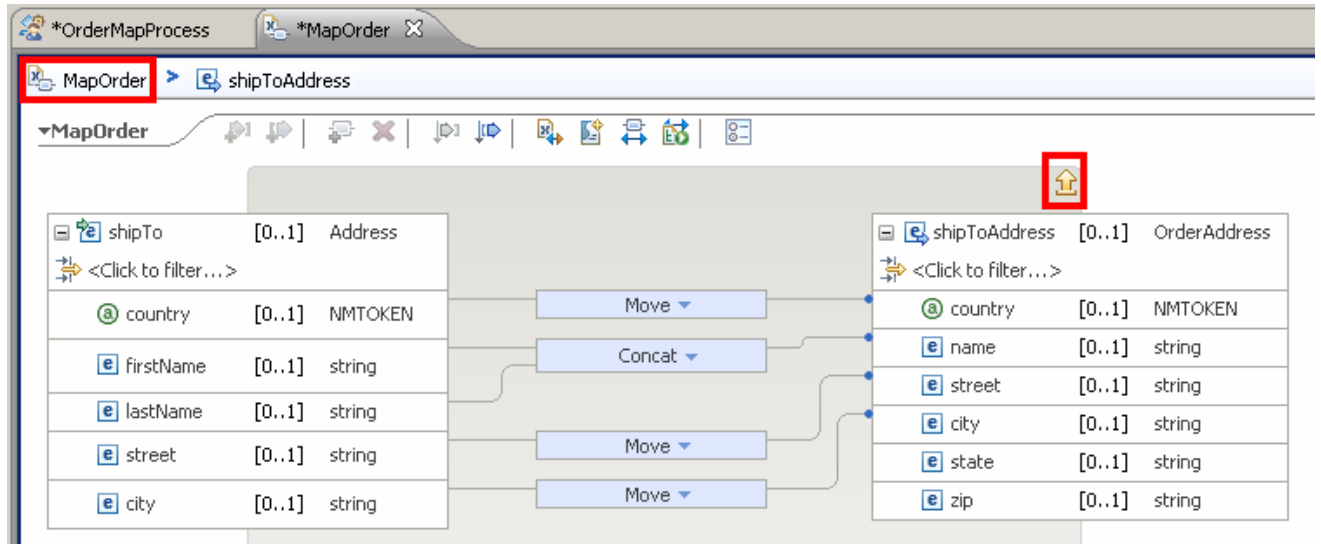


- e. To create the name in the target, you want it to be the first name followed by the last name. Therefore, you want to use the built-in concat (short for concatenate) transform. The mapping editor can automatically create a concat transform. The next step will show you how.

- f. Select the lastName element in the source artifact and drag the wire to the Move Transform. Notice that the Move transform automatically is changed to a Concat transform, and a small message flies from the transform to the upper right corner of the editor with information on what just changed



- ___ g. You are now finished with the If map. To get back to the main map, click the **yellow up arrow** above the target artifact in the map or use the **breadcrumb** locator at the top-left corner of the XML Mapping Editor.



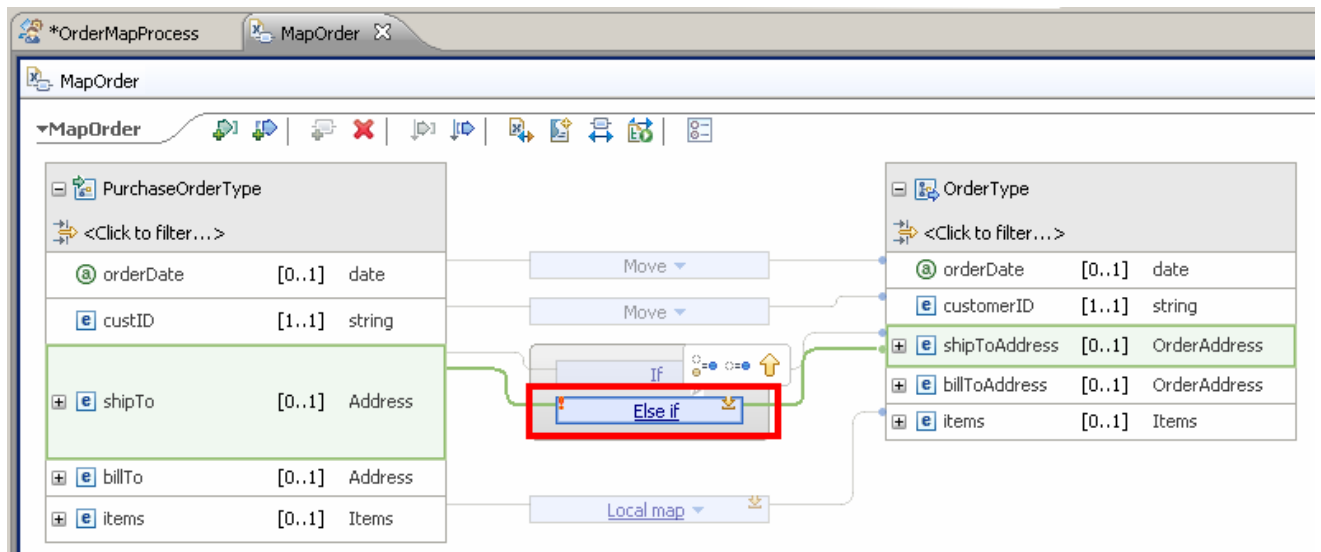
- ___ h. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

Part 4: Cut, Copy, and Paste transforms

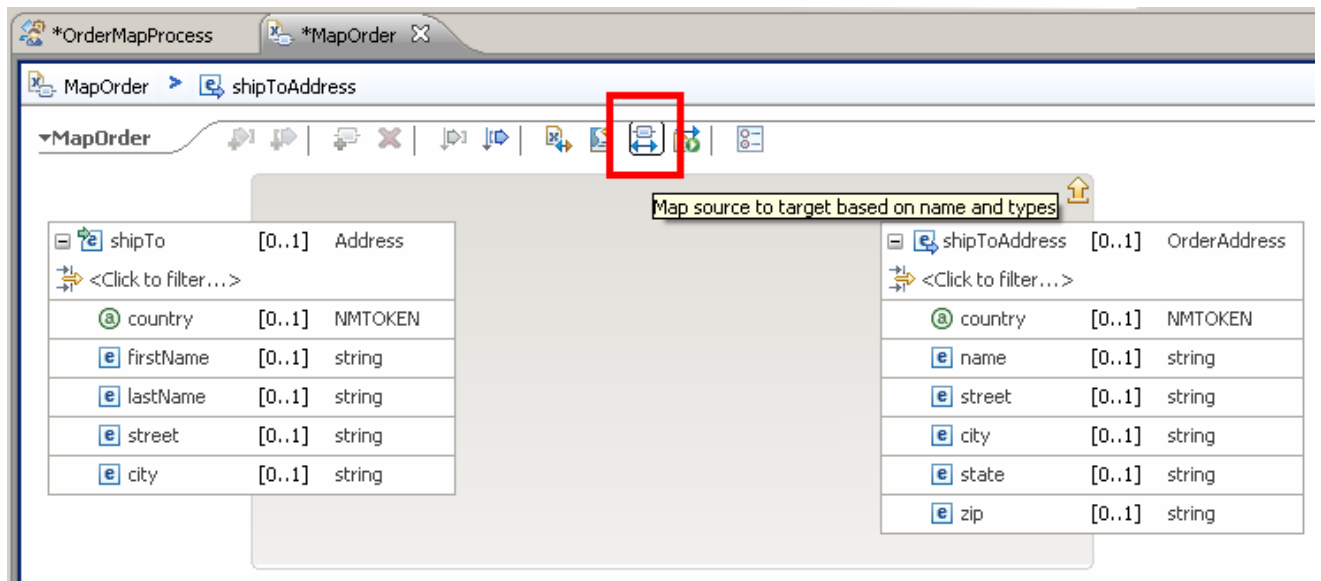
In this part of the lab, you will finish connecting the map inside the Else-If transform. To help you do this, you will copy the Concat transform inside the If map and paste it into the Else-if map. Version 7 XML Mapping Editor has the ability to cut, copy, and paste transforms. This option speeds development when reusing maps. For example, when building new maps from old maps you can copy transforms that took a bit of time to initially create and put them into a new map or submap. There is validation on the use of the paste function in that if the target schema does not match the source schema you copied, you will not be allowed to paste.

___ 1. Finish connecting the Else-If map using copy and paste transforms.

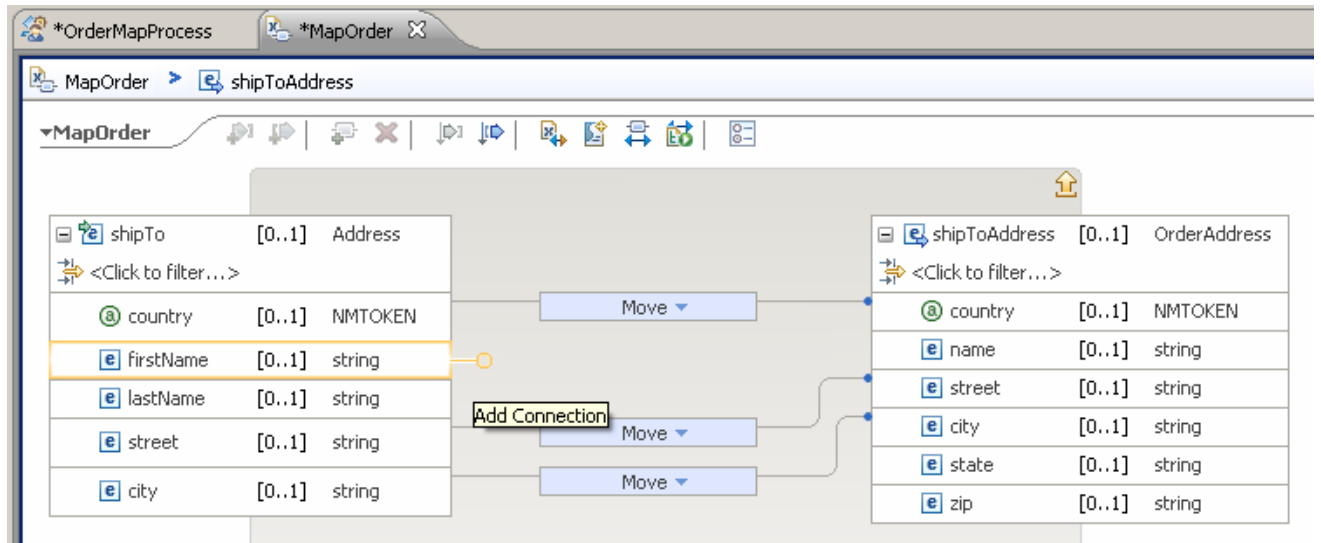
___ a. Click the Else if text (which is a link) or double click the Else if transform to open the map



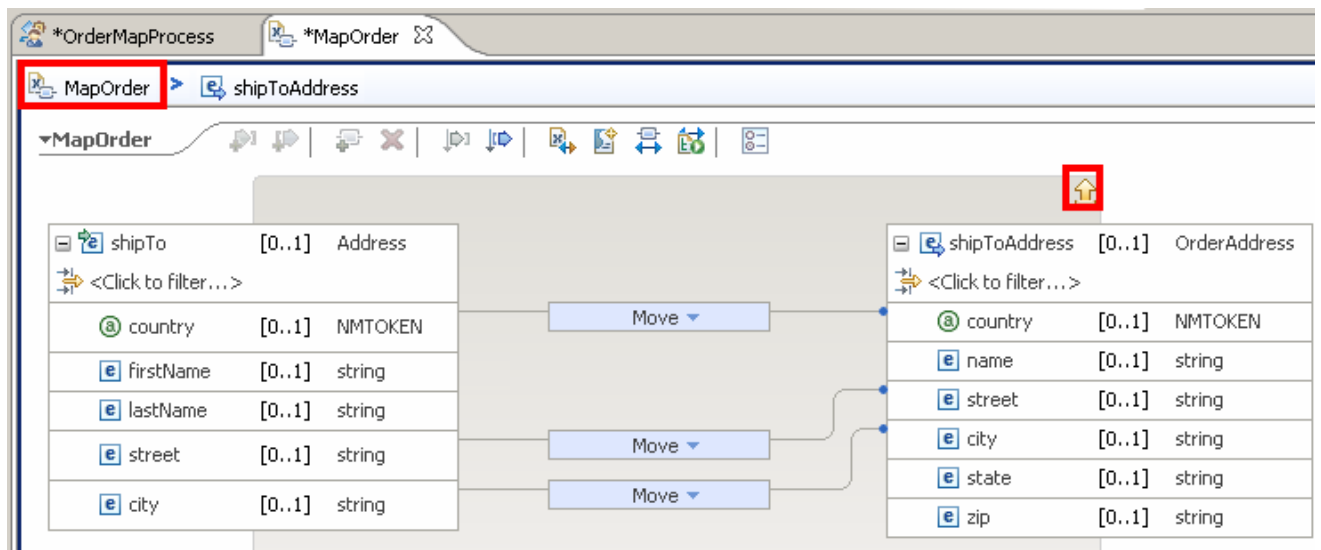
___ b. Given the source and target elements are named the same, click the **Map source to target based on name and types** button to automatically create transforms.



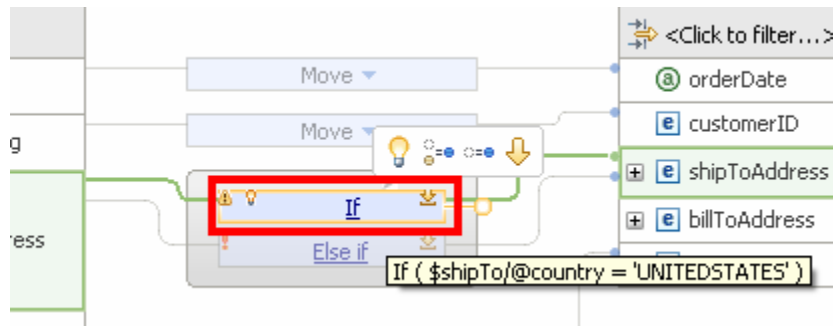
- c. Again, country, street, and city will be automatically mapped with a Move transform. FirstName, lastName, and name will not be automatically mapped since the names do not match.



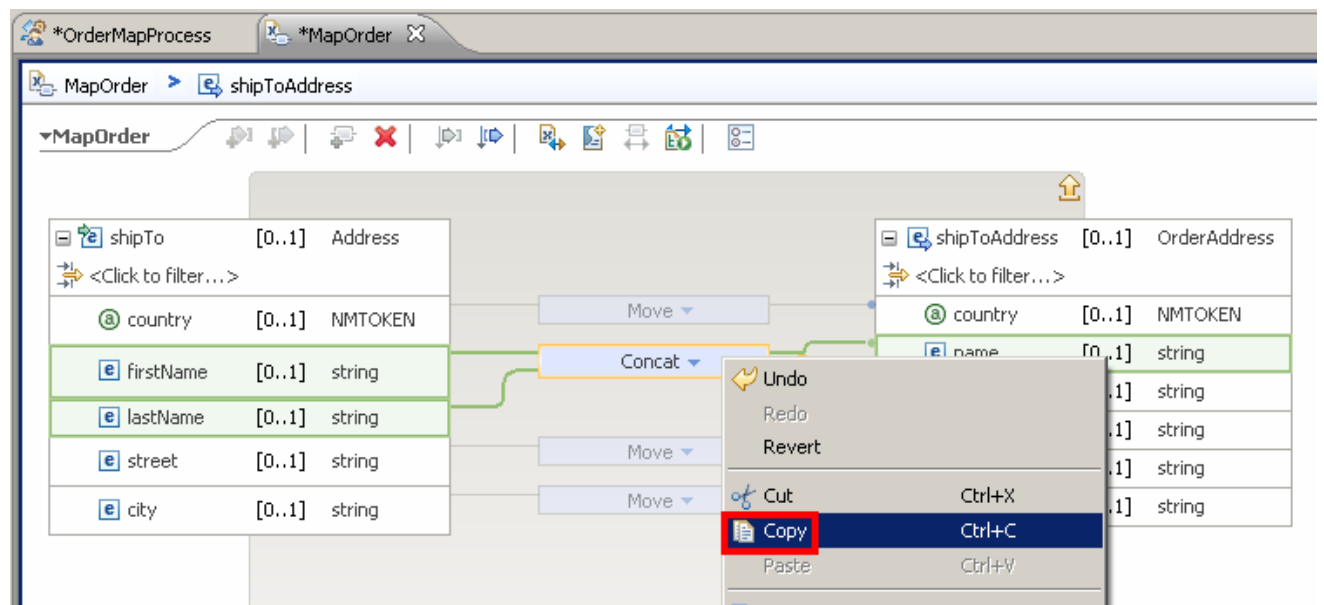
- d. Instead of manually creating a Concat transform for firstName and lastName into name, you will copy the Concat transform from the If map. You will need to open the If map again in order to copy the Concat transform. To get to the If map, go back to the main map by clicking the **yellow up arrow** above the target artifact in the map or use the **breadcrumb** locator at the top-left corner of the XML Mapping Editor.



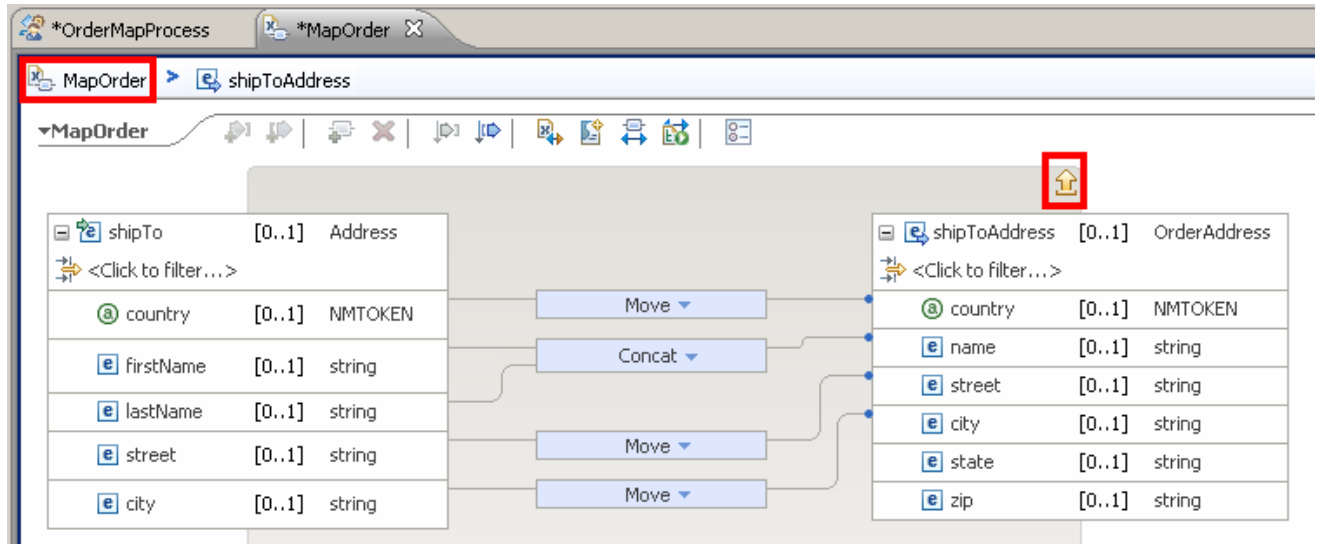
___ e. Once back on the main map, click the **If** text (which is a link) or **double click the If** transform to open the If map.



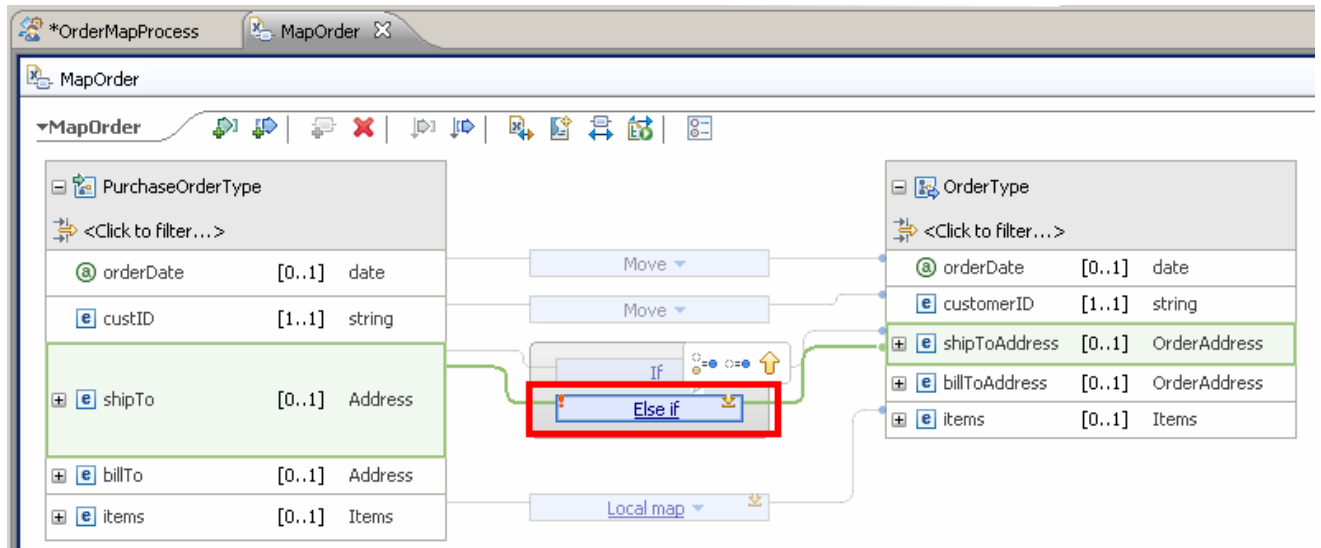
___ f. Once inside the If map, right click the Concat transform and click Copy.



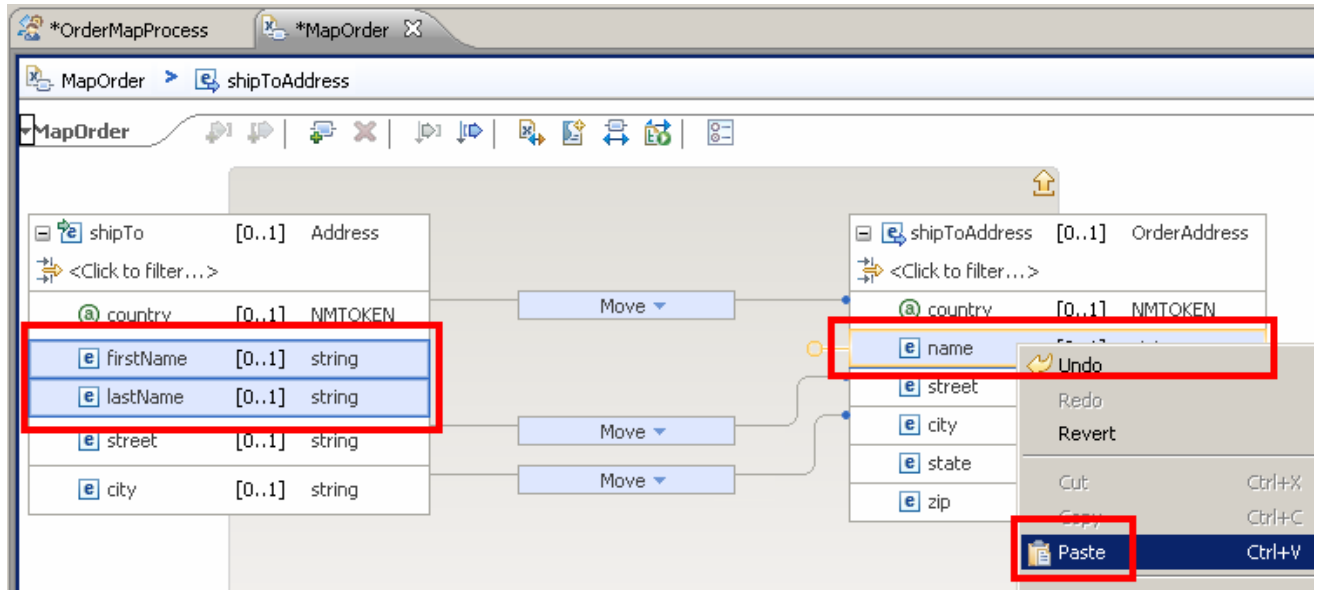
- g. Now that the transform is copied, go back into the Else if map. To get back to the main map, click the **yellow up arrow** above the target artifact in the map or use the **breadcrumb** locator at the top-left corner of the XML Mapping Editor.



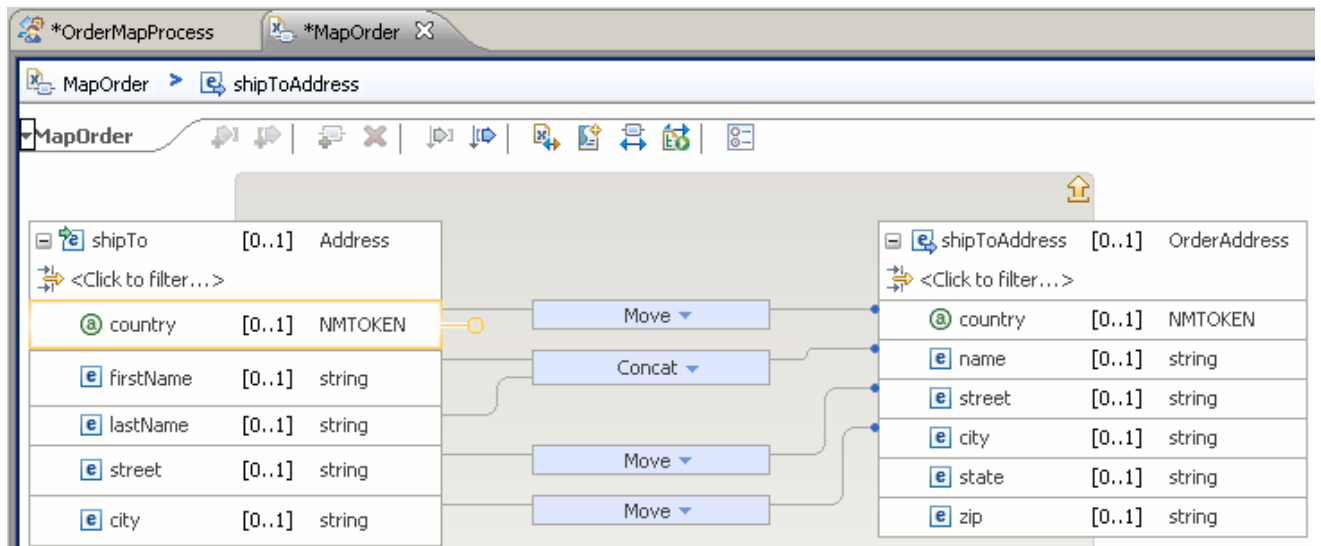
- h. Click the Else if text (which is a link) or double click the Else if transform to open the map.



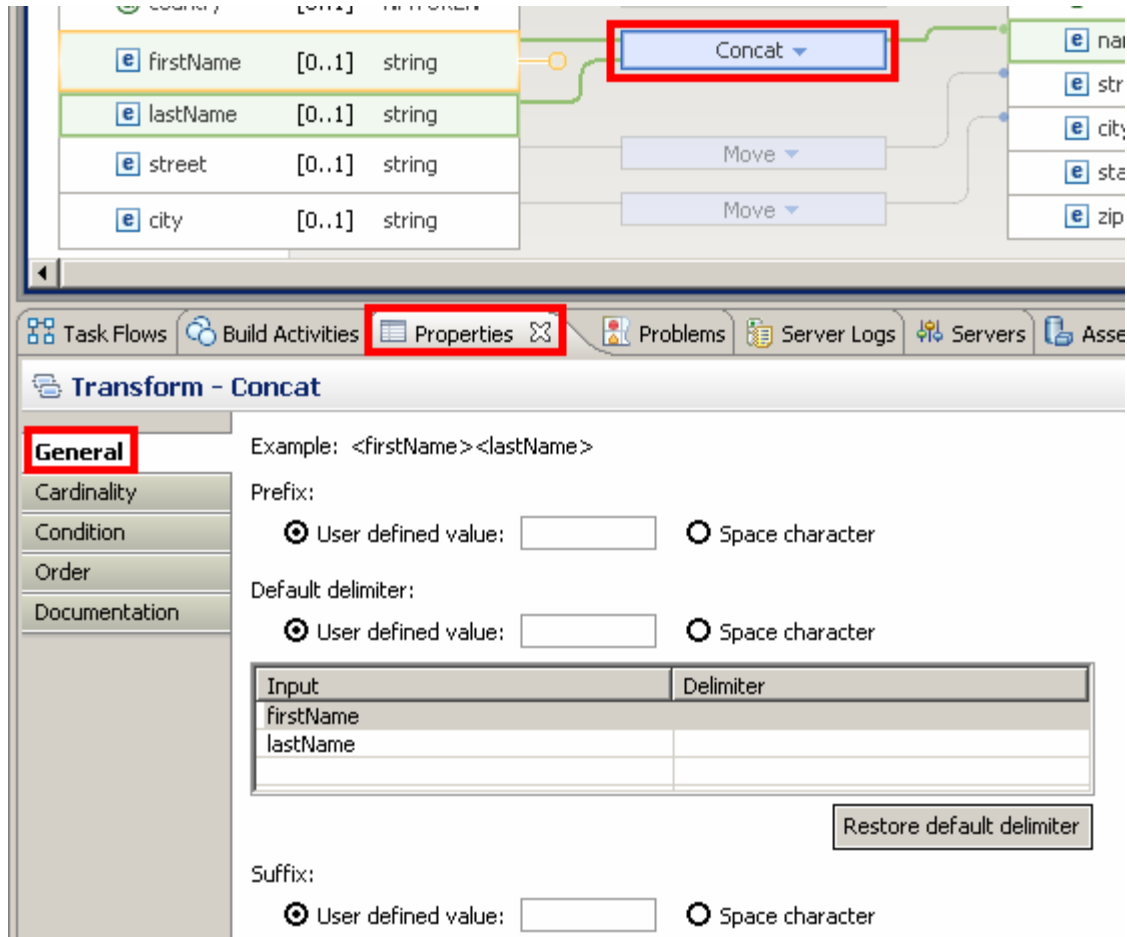
- i. Once in the Else if map, select the same input and output elements of the Concat transform by holding down the **Ctrl** button and selecting **firstName** and **lastName** elements in the source and the **name** element in the target. **Right click** any of the three highlighted elements and select **Paste**.



- j. After pasting you will notice that the Concat transform has been pasted and automatically wired to the inputs and outputs of the transform. Though copying and pasting the Concat transform is an easy example, you can imagine a complex transform that would take quite a while to build would be easy to cut, copy, and paste.



- ___ k. **Click the Concat transform** to bring focus and open the **Properties view** to the **General tab**. Notice that each transform has its own properties that can be edited and would be copied to the new location.



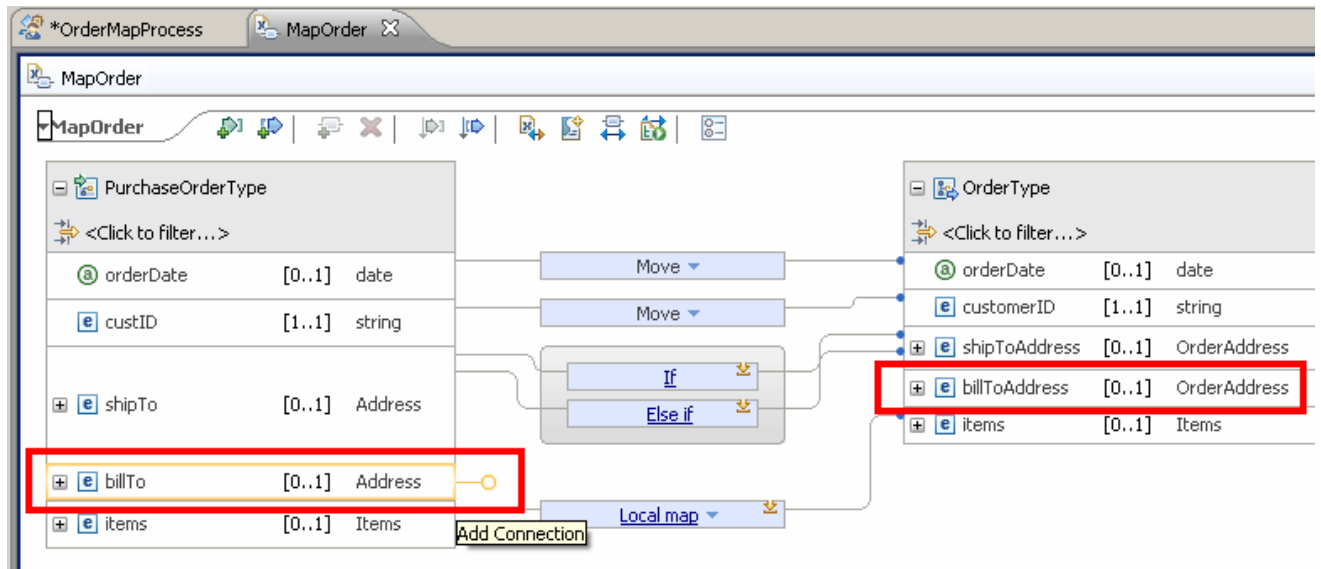
- ___ l. You are now finished with the Else if map. Go back to the main map by clicking the **yellow up arrow** above the target artifact in the map or use the **breadcrumb** locator at the top-left corner of the XML Mapping Editor.

- ___ m. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

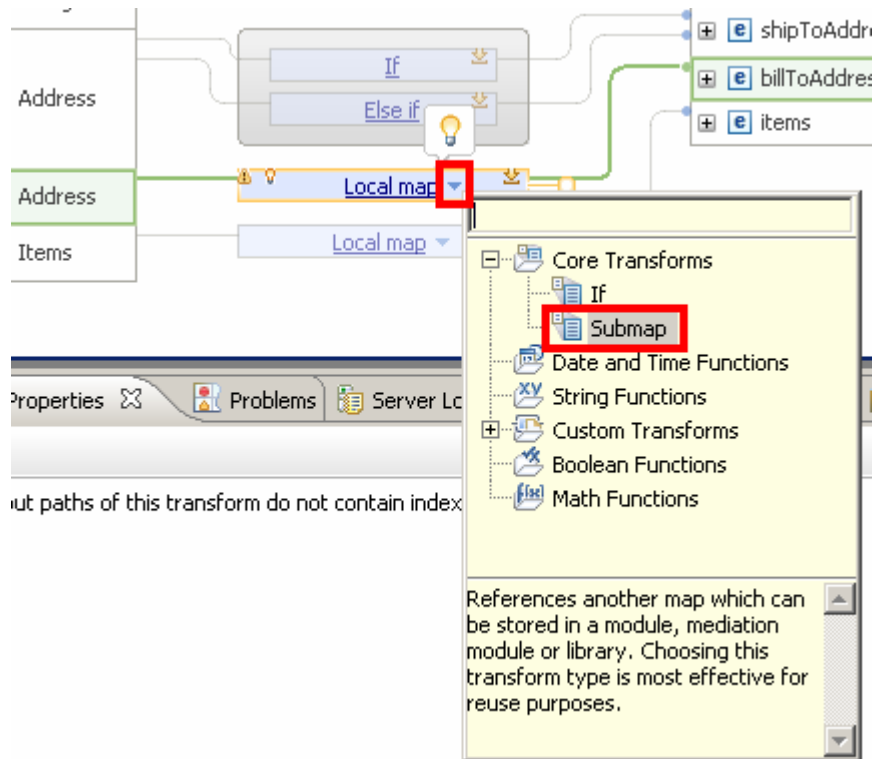
Part 5: Refactor to a submap

In this part of the lab, you will see how to refactor a local map into a submap. Version 7 brings the ability to quickly change a map transform into a submap. The reason why you would change a local map into a submap is to split up large maps into smaller, reusable pieces. Large maps can be hard to manage and not reusable. Therefore, using submaps makes map content reusable between several larger maps that may use the same local maps and can make large maps easier to manage.

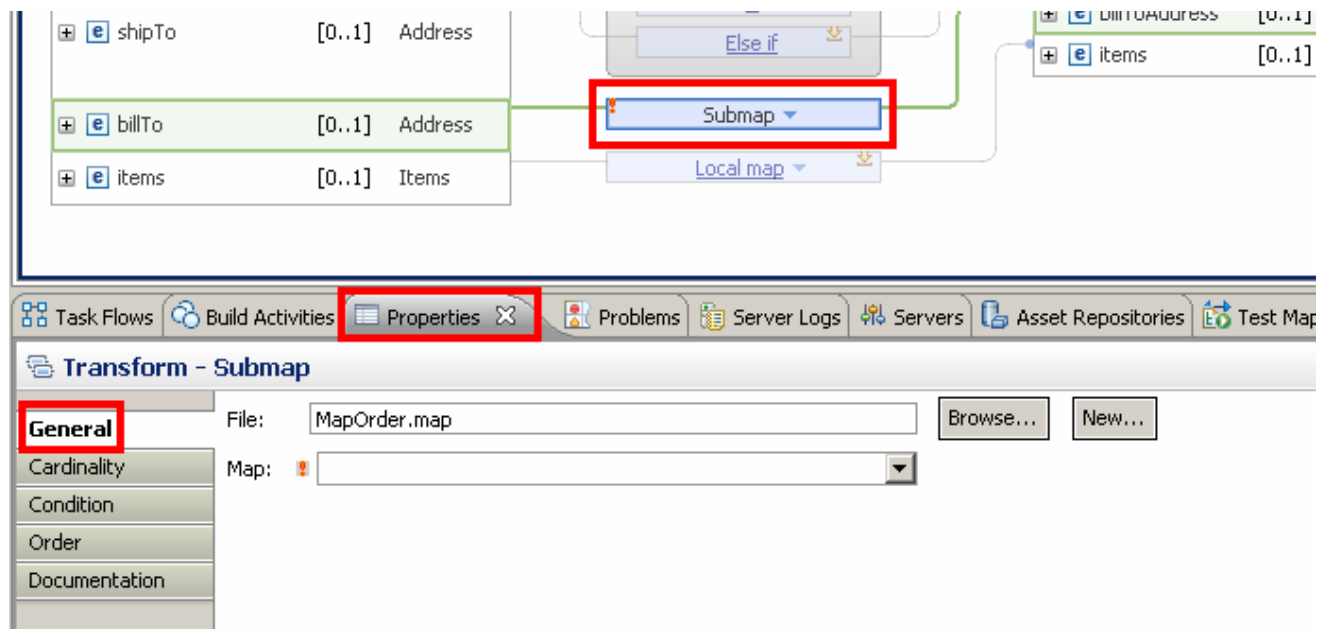
1. Create a local map and refactor it into a submap.
 - a. Select the **billTo** element on the source business artifact (left side) and drag a wire to **billToAddress** on the target business artifact (right side). A local map will be created automatically.



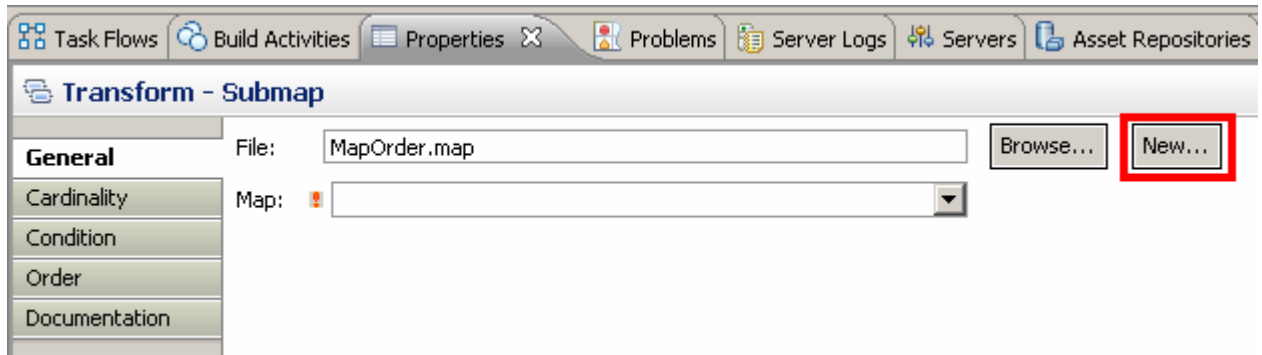
- ___ b. Click the **small blue triangle** to the right of Local map in the transform to bring up the built-in function menu. **Click the Submap option** in the built-in functions menu to change the Local Map transform to a Submap transform.



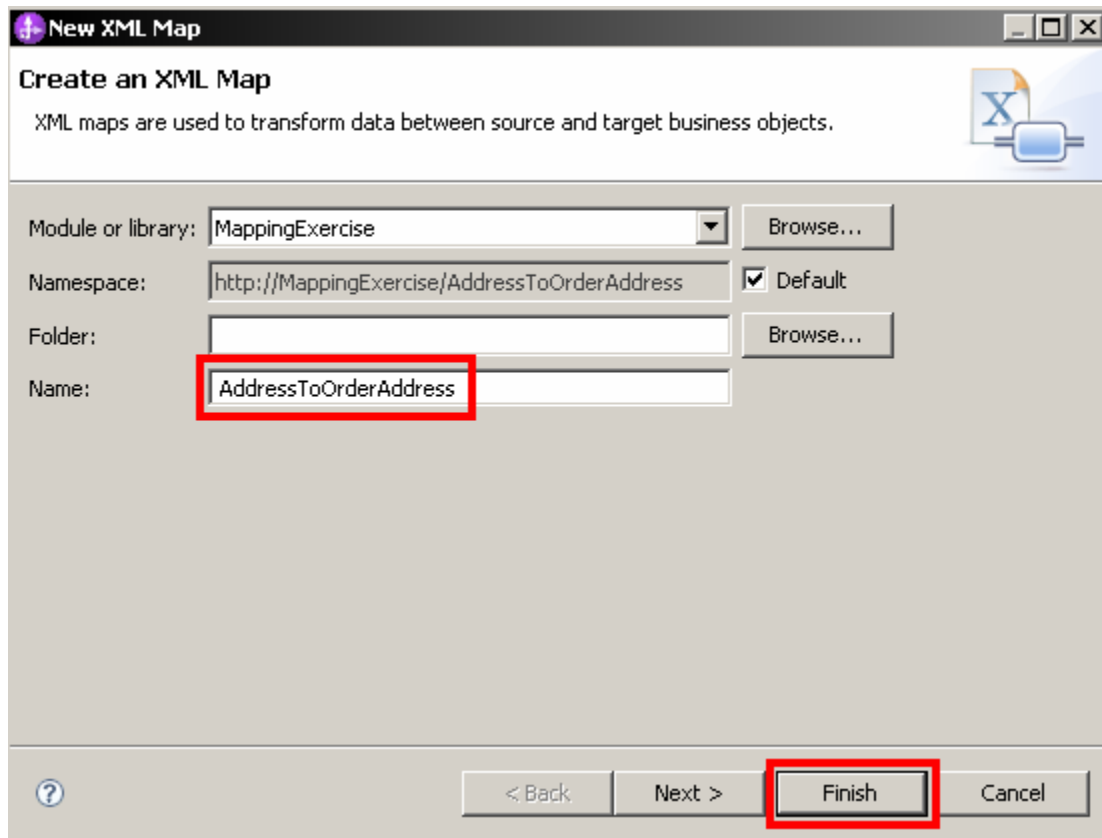
- ___ c. The transform will be changed to a Submap, but you will need to **identify the submap** in the **Properties view**, in the **General tab**.



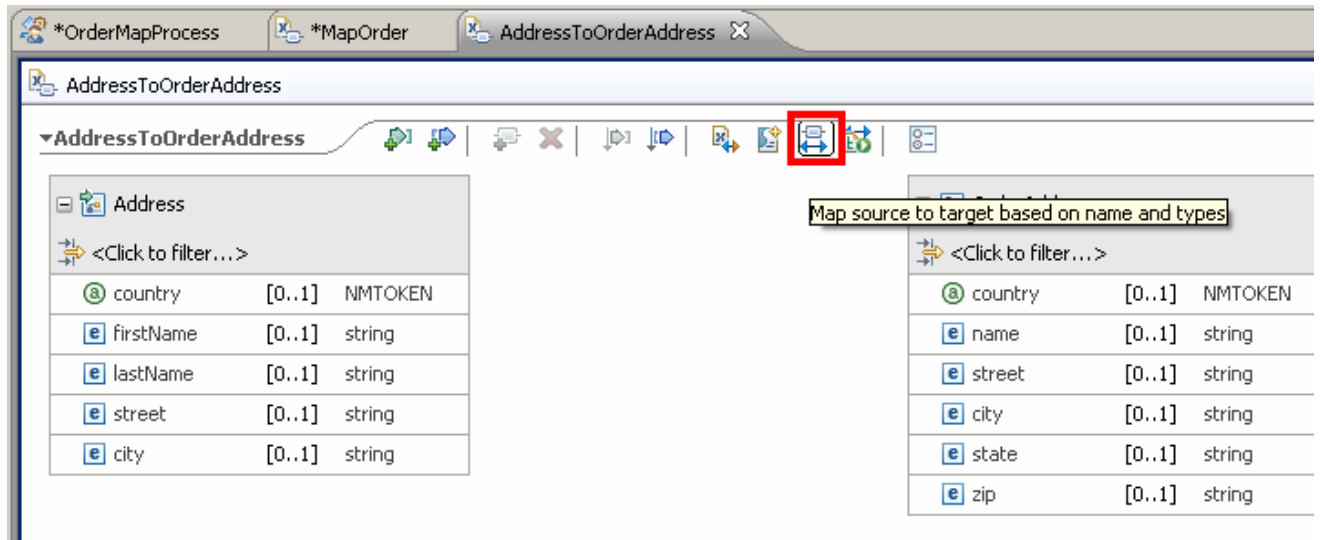
- 1) Click the New... button to have the XML Mapping Editor create a new submap based on the inputs and outputs that are already wired.



- d. In the new map wizard, you will notice that the map name is pre-filled in with the source and the target types that you are mapping. In this case, **AddressToOrderAddress**. Leave the submap name and click **Finish**.

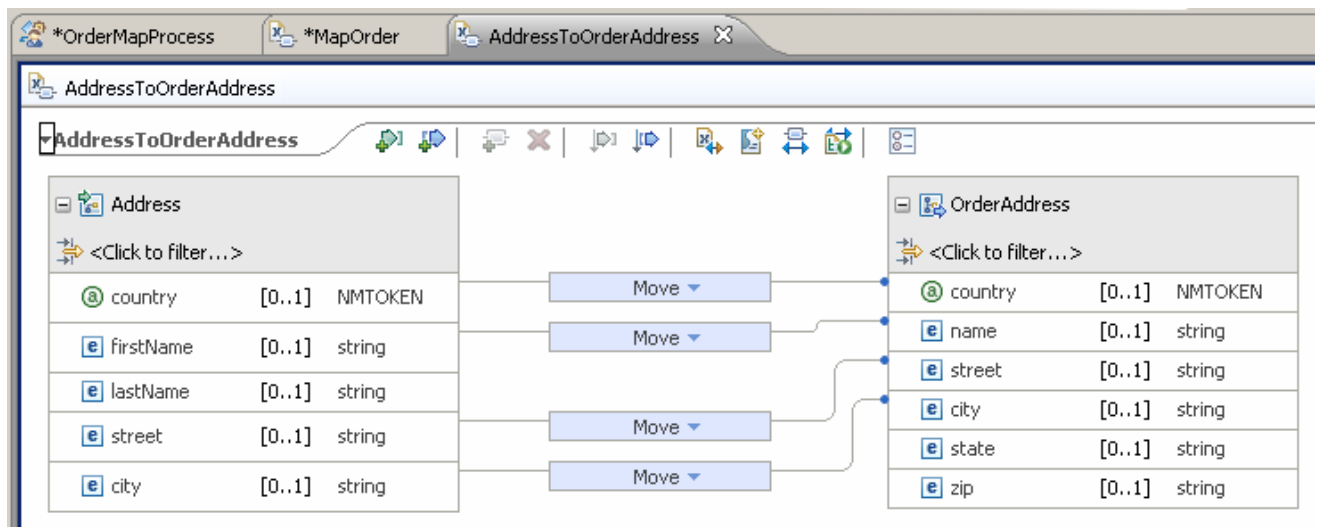


- e. The new map appears for you to connect source to target. Again, click the **Map source to target based on name and types** button.

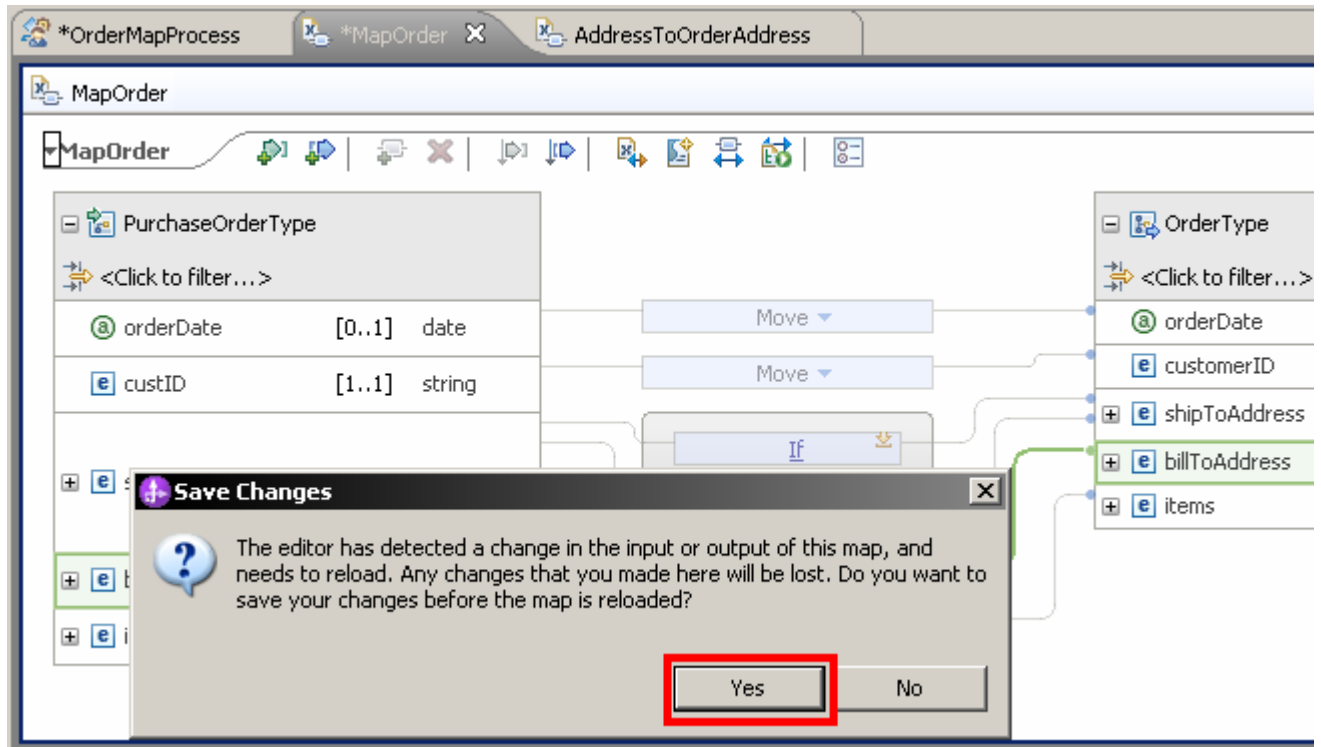


- f. This time, only connect firstName element from the source artifact (left side) to the name element in the target artifact (right side). Select the **firstName** element on the source business artifact (left side) and drag a wire to the **name** element on the target business artifact (right side). A move will be created automatically.

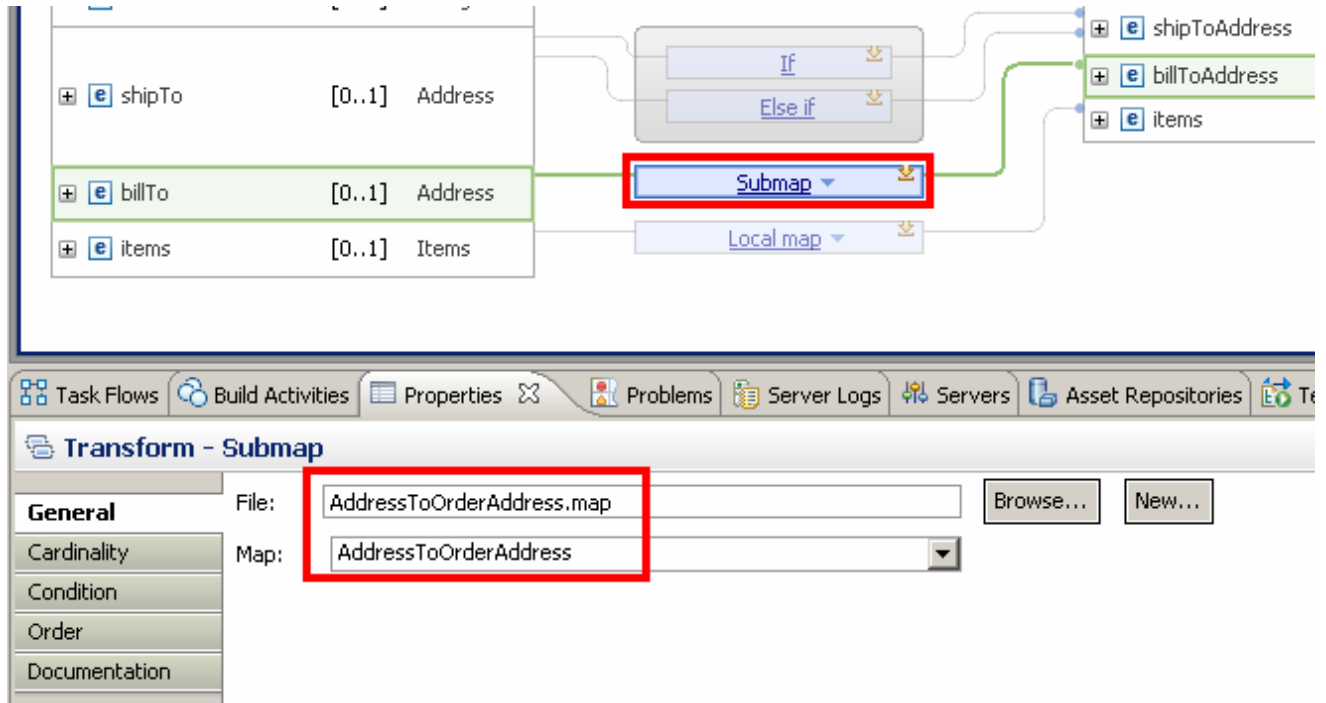
- g. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)



- h. Click the **MapOrder** tab in the XML Mapping Editor. You will be prompted that the XML Mapping Editor will save changes while reloading the map with the new found submap. **Click Yes.**



___ i. Once MapOrder has been refreshed, you should see the **Submap** transform is now pointing to **AddressToOrderAddress**.



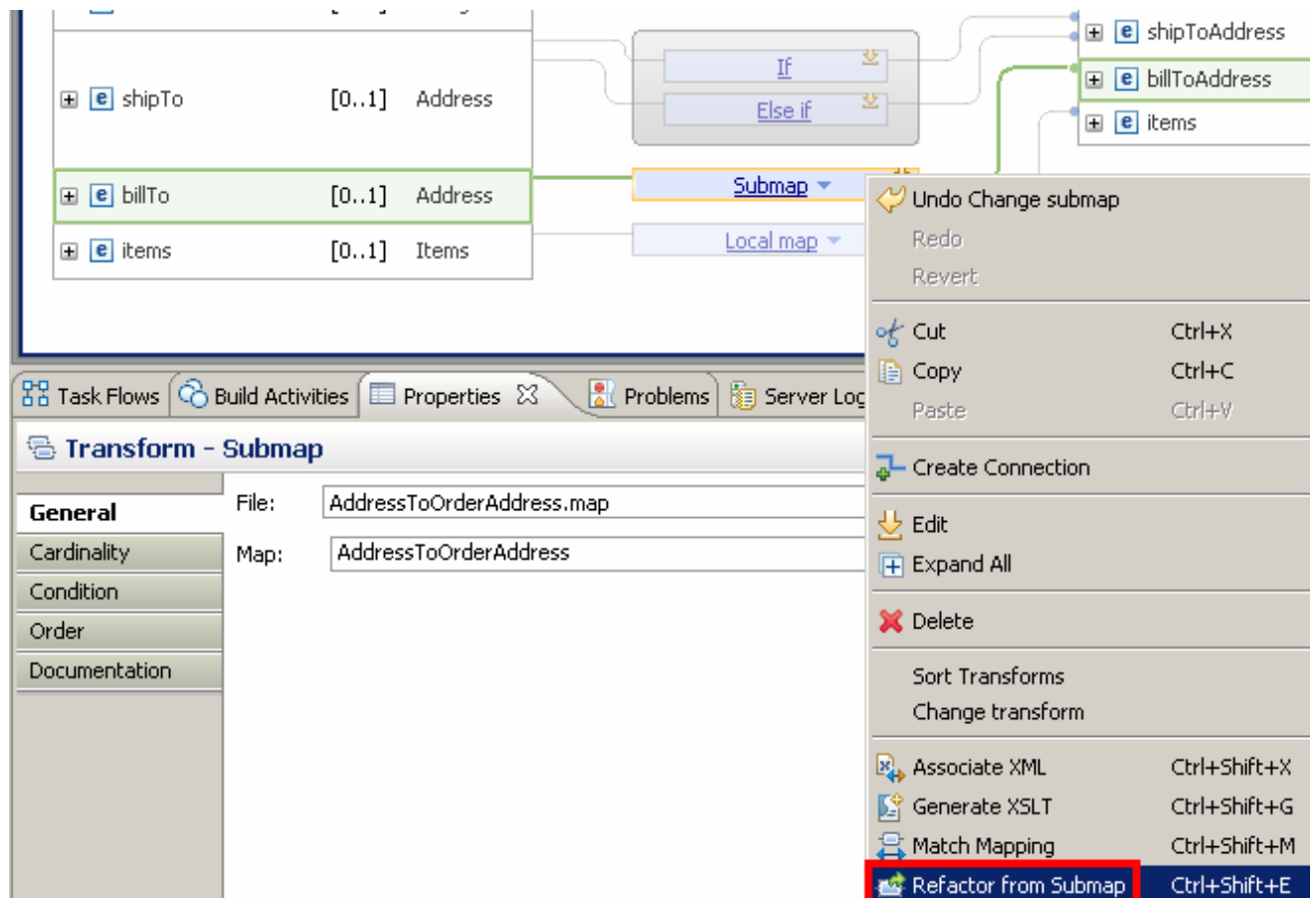
___ j. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

Part 6: Refactor from a submap

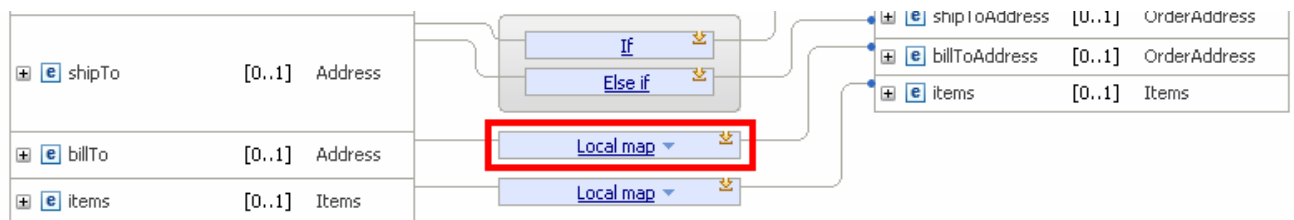
In this part of the lab, you will see how to refactor from a submap to a local map. Version 7 also brings the ability to quickly change to a local map from a submap. A Local map is created carrying along the containing mappings from the submap. This option helps if you ever want submaps to be brought into a larger main map.

___ 1. Refactor AddressToOrderAddress from a submap into a Local map.

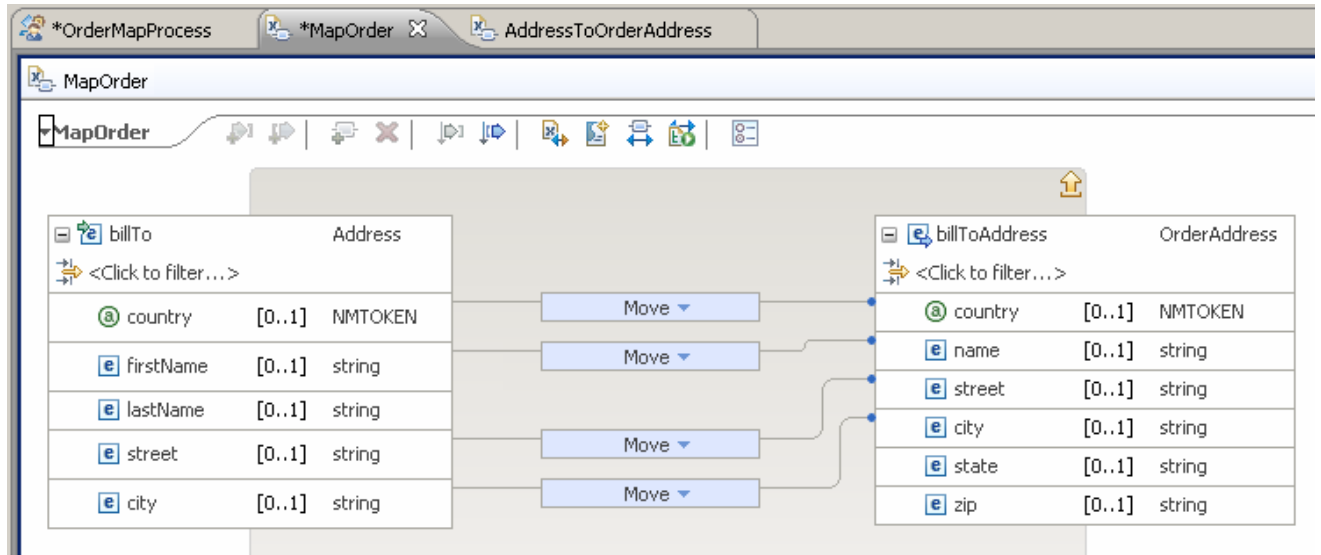
___ a. Right-click the **Submap** transform and select **Refactor from Submap**.



___ b. The submap is now changed back to a local map.



- c. **Double-click the Local** map to see that the contained mappings were propagated from the Submap into the Local map transform. Also notice that the map is still in MapOrder and not split out in the AddressToOrderAddress submap.



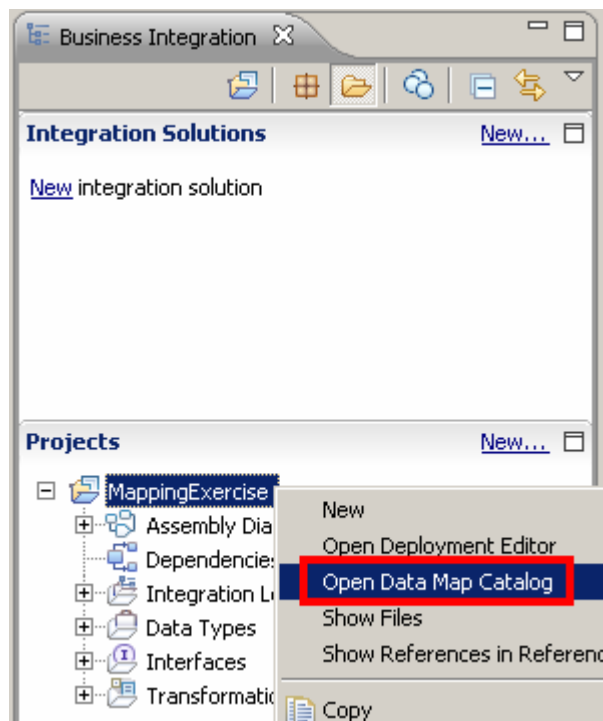
- d. Save the changes. From the menu select **File → Save** (or press **Ctrl + S**)

Part 7: Data map catalog

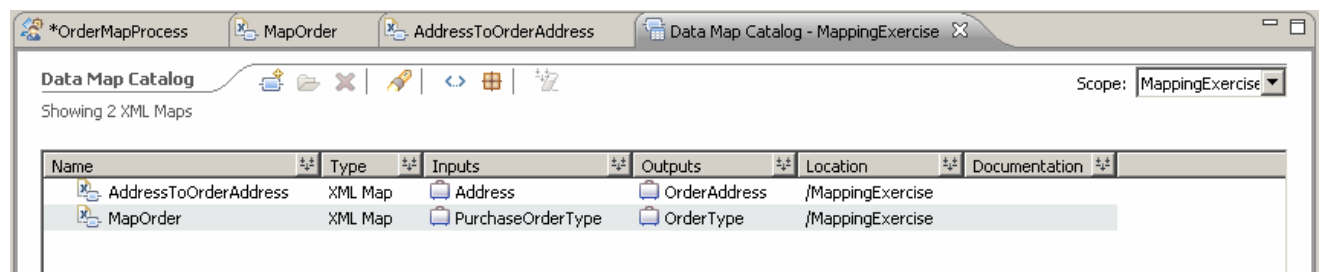
In this part of the lab, you will see the data maps you have in your module in the new data map catalog view. Previously to version 7, there was not a good way to see or work with all maps in solution, module, or library. WebSphere Integration Developer version 7 introduces a data map catalog in to make this possible. To help organize, you can filter rows and sort by column headers by name, type of maps, inputs, outputs, location, and documentation of the maps. There is also a text search for all the results. Wildcards are accepted when filtering to find a map or certain maps. Toggle buttons in the data map catalog toolbar allow you to show and hide content tags and namespaces. In addition, you can see other objects that use the map in the references view. Finally, you can also create a new data map from the data map catalog.

___ 1. Add

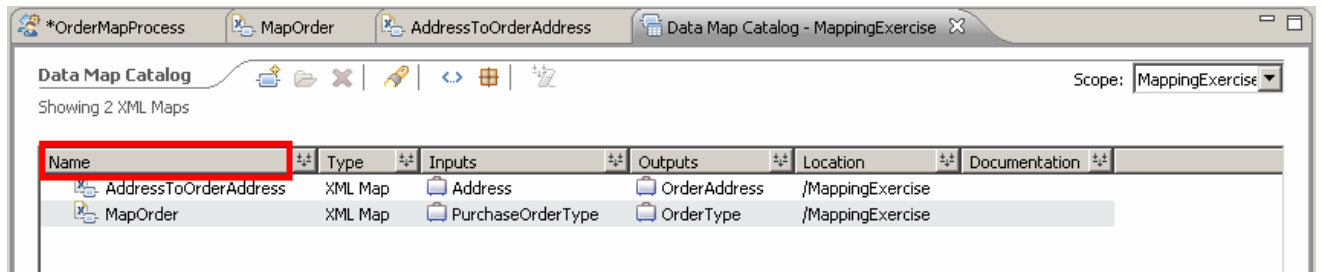
___ a. Right-click the **MappingExercise** module and select **Open Data Map Catalog**



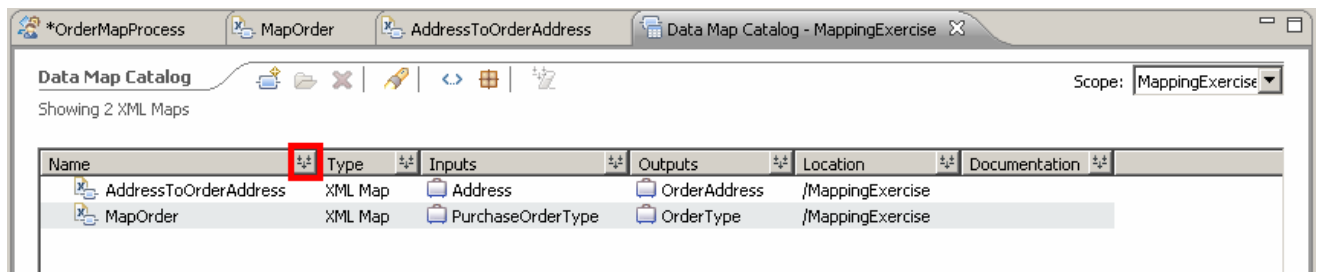
___ b. Though you only have two maps in this example module, realize that this viewer will be very valuable for modules, libraries, or entire workspaces that have many maps.



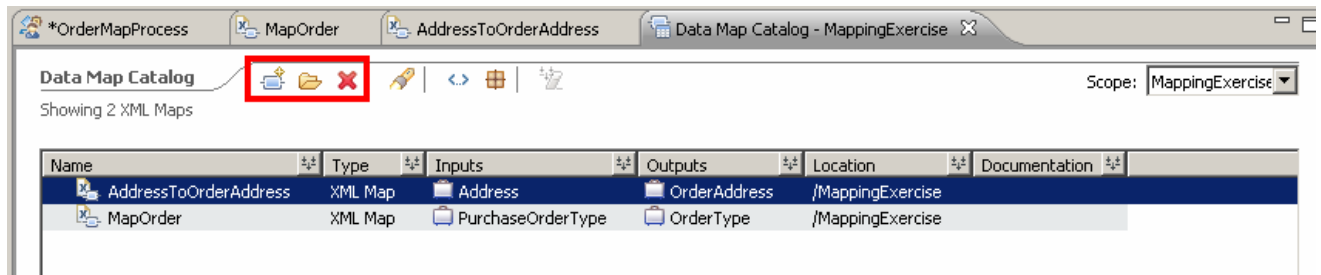
__ c. By clicking on the column headers you can quickly sort ascending or descending.



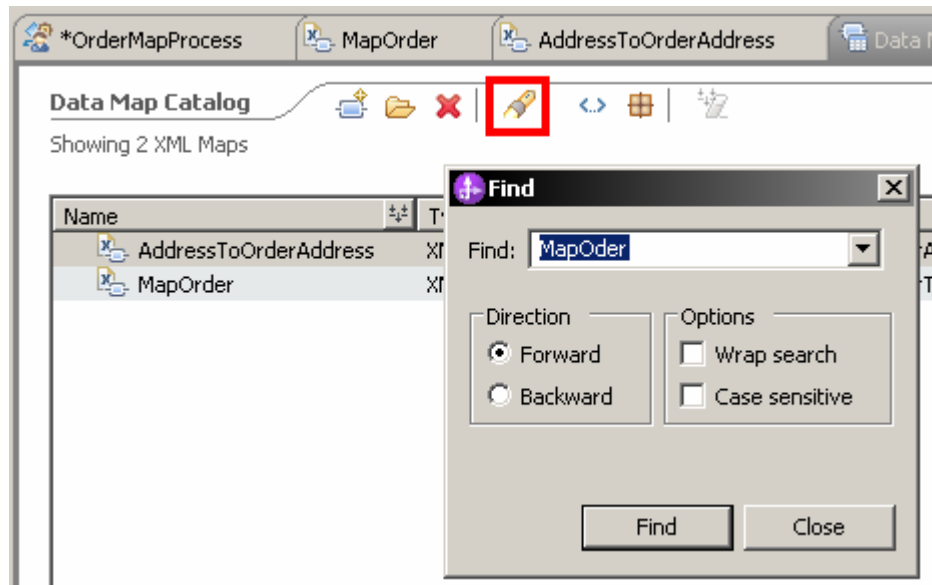
__ d. By clicking on the filter icon you can start typing filter text to limit the list. Notice the similarity between these filter options and the task list widgets in business space.



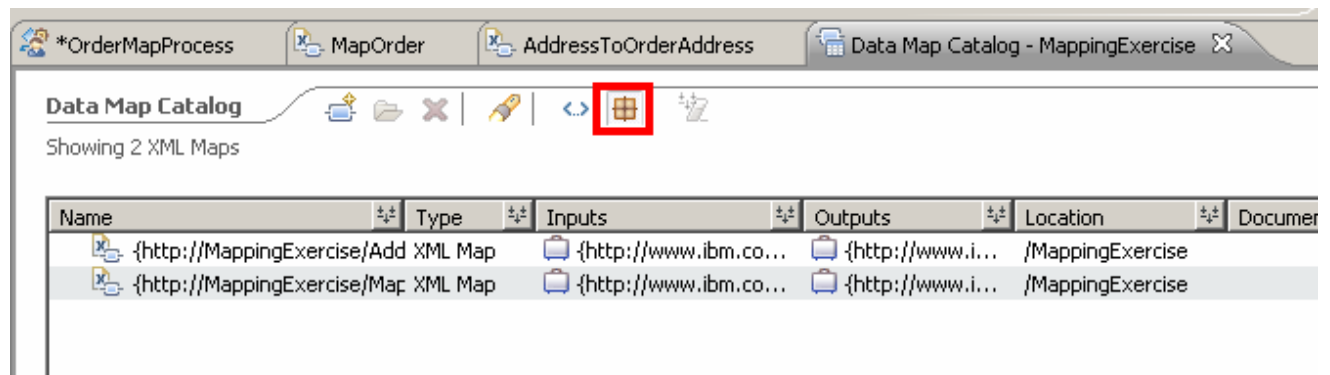
__ e. In the menu bar, you have the option of creating a data map from this viewer, opening the data map in the XML Mapping Editor, or deleting a map.



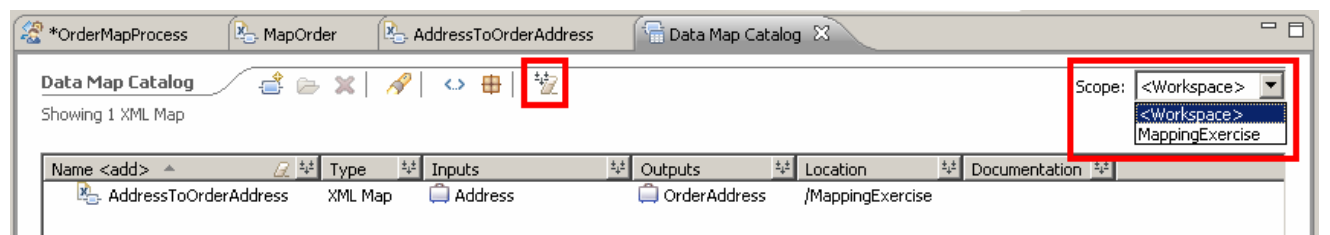
__ f. You can also find a data map using the toolbar Find icon.



__ g. You can show content tagging and namespace information



__ h. You can delete all column filters with one button and change the scope of the data map catalog to open or focus the view.



__ i. Save the changes. From the menu select **File** → **Save** (or press **Ctrl + S**)

What you did in this exercise

In this lab, you

- Imported a project interchange file into the WebSphere Integration Developer V7.0 development environment
- Created an data map
- Use a data map activity in BPEL
- Use conditional support like if, else if, and else conditions in an XML Map
- Identified the many built-in functions added for XML maps in version 7
- Copy and pasted a transform between maps
- Refactored to and from submaps in an XML Map
- Used the data map catalog to find maps faster and for an organized view of maps

Solution instructions

- ___ 1. Import **Solution** Project Interchange file
 - ___ a. With a blank workspace in WebSphere Integration Developer, Go to **File → Import → Project Interchange**
 - ___ b. Click the top **Browse** and navigate to **<LAB_FILES>/solution/WBPMv70_XMLMapLab_PI_Solution.zip**
 - ___ c. Select all projects listed
 - ___ d. Click **Finish** button
- ___ 2. Walk through lab to see how content was created