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	3
Part 4: Cut, Copy, and Paste transforms2	7
Part 5: Refactor to a submap	3
Part 6: Refactor from a submap	9
Part 7: Data map catalog4	1
What you did in this exercise4	4
Solution instructions	5

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></wid_home>	C:\IBM\WID70	/opt/IBM/WID70
<wte_home></wte_home>	C:\IBM\WID7_WTE\runtimes\bi_v7	/opt/IBM/WID7_WTE/runtimes/bi_v7
<wps_profile_home></wps_profile_home>	<wte_home>\profiles\qmwps</wte_home>	<wte_home>/profiles/qmwps</wte_home>
<lab_files></lab_files>	C:\Labfiles70\WID\Mapping	/tmp/Labfiles70/WID/Mapping
<workspace></workspace>	<lab_files>\workspace</lab_files>	<lab_files>/workspace</lab_files>
<solution></solution>	<lab_files>\workspace\solution</lab_files>	<lab_files>\workspace/solution</lab_files>

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></server_name>	server1	sssr011	
<was_home></was_home>	C:\Program Files\IBM\WebSphere\AppServer	/etc/sscell/AppServer	
<hostname></hostname>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<soap_port></soap_port>	8880	8880	
<telnet_port></telnet_port>	N/A	1023	
<profile_name></profile_name>	AppSrv01	Default	
<userid></userid>	N/A	Ssadmin	
<password></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>

- Workspace Launcher	×
Select a workspace IBM WebSphere Integration Developer 7.0 stores your projects in Choose a workspace folder to use for this session.	a folder called a workspace.
Workspace: C:\LabFiles70\WID\Mapping\workspace	Browse
Use this as the default and do not ask again	
	OK Cancel

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

😨 Business Inte	gration 🛛		' 🗆
	B 🖶 🗄	🗆 🔄	~ ~
Integration So	lutions	New	
New integration :	solution		
	New	•	
	🔁 Import		
	🛃 Export 🕅		
Projects	🔊 Refresh	-	
<u>New</u> business int	Add to Asset Repository	y	
	Properties		

____b. From the Import, dialog, expand Other and select Project Interchange from the tree

🌐 Import	
Select	
Import a project and its dependent projects from	n a Zip file.
Select an import source:	
type filter text	
Web Web services XML Other FTP Localized Model Project Interchange	
? < <u>B</u> ack <u>N</u> ext	> Einish Cancel

- ___ 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.

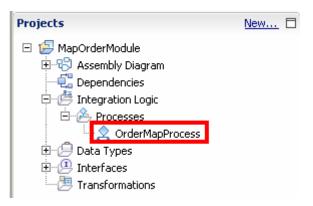
🚯 Import Project In	terchange Conten	ts		_ 🗆 🗙
Import Projects Import Projects from a	a zip file.			
From zip file: Project location root:				Browse
Select All Deselec	t All Select Referen	nced		
0	< Back	Next >	Finish	Cancel

- ____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.

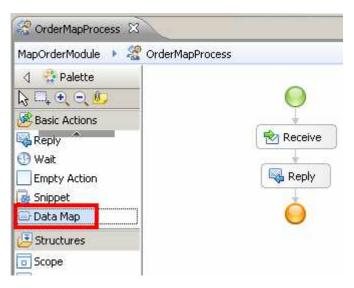
Part 2: Add data map to BPEL flow

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.

🌐 New Data Map				
Select a Data Map Type Data maps are used to transform data between source and target business objects.				
	Use an XML map for most transformations. An XML map uses standard XSLT for data transformation.			
O Business object map	Use a business object map only when you need to perform specific types of transformations. <u>More</u>			
0	< Back Next > Einish Cancel			

- ____ d. Click Next.
- ____e. Type **MapOrder** in for the name of the map.

🚯 New Data Map			
Create an XML XML maps are use	. Map ed to transform data between source and target busin	ess objects.	X
Module or library: Name <u>s</u> pace: F <u>o</u> lder: N <u>a</u> me:	MapOrderModule http://MapOrder MapOrder	Browse	New
0	< <u>B</u> ack <u>N</u> ext >	Einish	Cancel

___ f. Click Next.

____g. Click the Add Variable button next to the Input Variables area

🥵 New Data Map	
Select Input and Output Business Objects The XML map will transform the input business object into the output business object.	X
Input Variables:	
	Add Variable Remove Variable
Output Variables:	
	Add Variable Remove Variable
⑦ < Back Next > Finish	Cancel

___h. Select the **input1** variable and click **OK**.

🚯 Select a Variable		_ 🗆 ×
Variable Name (? = any	character, * = a	any String):
Matches:		
😑 input1		
output1		
Variable Type:		
PurchaseOrderType		
0	ОК	Cancel

____i. Click the Add Variable button next to the Output Variables area

🕀 New Data Map	_ □ ×
Select Input and Output Business Objects The XML map will transform the input business object into the output business object.	X
Input Variables:	
input1 http://www.ibm.com/po - PurchaseOrderType	Add Variable
	Remove Variable
Output Variables:	
	Add Variable
	Remove Variable
Image: Second	Cancel

____j. Select the **output1** variable and click **OK**.

🕹 Select a Variable 📃 🗆 🗙
Variable Name (? = any character, $*$ = any String):
Matches:
e input1
Variable Type:
OrderType
OK Cancel

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

🦺 New Data Map				
Select Input and Output Business The XML map will transform the input busine		he output busine:	ss object.	X
Input Variables:				
input1 http://www.ibm.com/po - Purc	haseOrderType:			dd Variable emove Variable
, Output Variables:				
output1 http://www.ibm.com/order -	OrderType		<u></u>	dd Variable emove Variable
0	< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

I. 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.

😤 *OrderMapProcess	AapOr	der 🛛							
Re- MapOrder									
▼MapOrder 🛛 🎝	1 🕼 🕾	X]Þ1	100 B	1 🛱 🔁	0- 0-				
🖃 🚵 PurchaseOrderT	уре			Map so	ource to target I	based on	name and types		
⇒ → <click filter="" to=""></click>							Click to filter>		
(a) orderDate	[01]	date]				(a) orderDate	[01]	date
e custID	[11]	string]				e customerID	[11]	string
🕀 🖻 shipTo	[01]	Address					€ shipToAddress €	[01]	OrderAddress
🛨 🖻 billTo	[01]	Address	1				🗄 🖻 billToAddress	[01]	OrderAddress
🗄 🖻 items	[01]	Items	1				🛨 🖻 items	[01]	Items
			1			L			

____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.

- MapOrder								
MapOrder 🏾 🎜	a 🎲 🕾	X [D]	î⊳	🗣 🖻 🛱 👪 📴				
🖃 🗽 PurchaseOrderT	уре]			🖃 🔣 OrderType		
⇒ <click filter="" to=""></click>						⇒ <click filter="" to=""></click>		
(a) orderDate	[01]	date		Move 🔻]	(a) orderDate	[01]	date
e custID	[11]	string	-			e customerID	[11]	string
	[01]	Address	-				[01]	OrderAddres:
E billTo	[01]	Address				🗉 🖻 billToAddress	[01]	OrderAddres:
• e items	[01]	Theme		Local map 👻 💆	<u> </u>	🗄 🖻 items	[01]	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).

😤 *OrderMapProcess 🛛 🖳 *MapOrder 🛛									
RapOrder									
-MapOrder 🔊 🌮 🎏 🗶 🗈	🕪 🕵 😫 🛍 💷								
🖃 😰 PurchaseOrderType		🖃 💽 OrderType							
→ <click filter="" to=""></click>		$\stackrel{1}{\rightarrow 1}{\rightarrow}$ <click filter="" to=""></click>							
@ orderDate [01] date	Move 🔻	(a) orderDate [01] date							
custID [11] string	Move -	e customerID [11] string							
	-								
€ billTo [01] Address									
■ e items [01] Items	Local map 👻 🗠								

____ o. Save the changes. From the menu select File \rightarrow 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.

😤 *OrderMapProcess 🛛 🖳 *MapOrder 🔀									
Re. MapOrder									
-MapOrder 🛛 🔊 🌮 🍃	🗙 jos jo) 🔖 🗳 🛱 👪 🖾							
🖃 ն PurchaseOrderType				🖃 💽 OrderType					
🚔 <click filter="" to=""></click>				\Rightarrow <click filter="" to=""></click>					
(a) orderDate [01]	date	Move 🔻		(a) orderDate	[01]	date			
		Move -		e customerID	[11]	string			
		leadean −	<u>v</u>	🗄 🖻 shipToAddress	[01]	OrderAddress			
	Address	^{® V} Local map ▼		🗄 🖻 billToAddress	[01]	OrderAddress			
🕒 🖻 billo 🛛 [U1]	Address			표 🖻 items	[01]	Items			
	Items	Local map 🔻	*						

- 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.

∃ 🔚 PurchaseOrder 券 <click filter="" to=""></click>			□ □ □ OrderType ⇒ <click filter="" to=""></click>	
 a) orderDate 	[01] date	Move *	(a) orderDate	[01] date
e custID	[11] string	Move -	e customerID	[11] string
7455003030 MARKANANANAN	1728/08 7 17/09/98	🕴 🗘 Local mar 😴 😤 🚽	🗄 🖻 shipToAddress	[01] OrderAddress
🛛 🖻 shipTo	[01] Address			[01] OrderAddress
🛿 🖲 billTo	[01] Address	Local map	ansforms	[01] Items
		El - El Custom	d Time Functions unctions Transforms Functions	

- _____ 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.

MapOrder	raj 🗶 🖧 🏟 K	P & 2 4 5 2		-
□ PurchaseOrder ⇒ <click filter="" to=""></click>			□ I Crick to filter>	
(3) orderDate	[01] date	Move =	(a) orderDate [01] date	
e custID	[11] string	Move - See cee	e customerID [11] string	
		L I I V	Te shipToAddress [01] OrderAddress	
🗄 🖻 shipTo	[01] Address		I I I I I I I I I I I I I I I I I I I	
🕀 🖲 billTo	[01] Address			1
E items	[01] Items	Local map * 2		-
Task Flows 🔗 Build a	Activities 🔲 Properties 🗄	🗙 📲 Problems 🚺 Server Logs	🕫 Servers 🕻 Asset Repositories	~
🗟 Transform - If	-			
Cardinality	input element is evaluated	against the condition. If the condition	evaluates to true, the transform is applied to the input elemen	t.
Condition				Edit
Order				

_____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.

*OrderMapProcess	🐁 *MapOrder 🛛			₽ [
🖳 MapOrder				
▼MapOrder 🔊	에 데 X 쿡 액	🗣 🗳 🛱 👪 🖾		
🖃 🕍 PurchaseOrderTyp	e		🖃 🔛 OrderType	
🔆 <click filter="" to=""></click>			🔆 <click filter="" to=""></click>	
(a) orderDate	[01] date	Move 🕶	 a orderDate 	01] date
e custID		Move - Cre ore	e customerID [11] string
	🥵 XPath Expression Buil	der	×	.1] OrderAddress
🕀 e shipTo	Build an XPath Expre	ssion	11	.1] OrderAddress
🗄 🖻 billTo		below with the help of content assist. I	Jse "Insert Simple	.1] Items
🕀 🖻 items	XPath" if you need more g	uidance,	1.1	
× [Insert Simple XPath			
器 Task Flows 🙆 Build Act	<type an="" expression<="" td="" xpath=""><td>here. Content assist available (Ctrl+sp</td><td>pace)></td><td></td></type>	here. Content assist available (Ctrl+sp	pace)>	
🕾 Transform - If			*	
Cardinality The in	Content assist available (Ctr	H+space)		d to the input element.
Condition	0		OK Cancel	Edit
Order				
Documentation				

____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.

🚯 Simple XPath Expre	ssion Builder				×				
Build a simple XPath Expression									
Select a field and optionally add filters and conditions.									
Expression:									
\$shipTo/@country =	'UNITEDSTATES'								
Select a field:									
<type filter="" to=""></type>					•				
Data Types Data Types Data Types Outry Street : Country Count	e : string string ing								
Array Fields	Where	Operator	Value	and/or (optional)	Delete				
Example	- Jacob								
 Add an optional co 	Value								
Operator =	UNITEDS	TATES			Delete				
Example									
<u>examplern</u>									
0				ок	Cancel				

____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.**

🚯 XPath Expression Builder		X
Build an XPath Expression		1
Type an XPath expression below with the help of content assis XPath" if you need more guidance.	it. Use "Insert Simple	>>
Insert Simple XPath		
⁰ \$shipTo/@country = 'UNITEDSTATES'		_
		7
Content assist available (Ctrl+space)		
0	ОК	Cancel

- ____j. Save the changes. From the menu select **File** \rightarrow **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

2	*OrderMapProcess	🌯 *MapOr	der 🛛										
	ApOrder												
E	🔆 <click filter="" to=""></click>						÷∏∓	< Click to filter>					
L	(a) orderDate	[01]	date		Move 🔻			(a) orderDate	[01]	date			
L	e custID	[11]	strina		Move 🥆 💡 🌅 ==			e customerID	[11]	string			
L				<u>a v</u>	If - Z		ŧ	e shipToAddress	[01]	OrderAddress			
L	🕀 🖻 shipTo	[01]	Address		Add	Else if	±	e billToAddress	[01]	OrderAddress			
L	🗉 🖻 billTo	[01]	Address			5	•	e items	[01]	Items			
L	표 💼 items	[01]	Items		Local map 🔻 👋								
				1									

	CUSTID	[11] string	E e shipToAddress	
	🗈 🖻 shipTo	[01] Address	Else if ■ e billToAddress [01] OrderAddress • e items [01] Items	
	🗉 🖻 billTo	[01] Address		
	🛨 🖻 items	[01] Items	Local map 🔻 💆	
	_			-
	Task Flows 🙆 Buil	ld Activities 🔲 Properties 🔀	🔹 🔝 Problems) 👸 Server Logs 🖗 Servers 🕼 Asset Repositories 🔂 Test Map 🛛 🗸 🖙	,
R	·		🔹 🕅 Problems 👔 Server Logs 🖗 Servers 🔓 Asset Repositories 🔂 Test Map 🛛 🗢 🗖	, <u> </u>
	Task Flows 🙆 Build	se if	Image: Server Logs Image: Se	ì E

____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.

*OrderMapProcess	🕾 *MapOrder 🕱				- [
🖳 MapOrder					
■MapOrder	10 i 10 10 10 10 10 10 10 10 10 10 10 10 10	> 🖳 😰 🛱 🚮 🖾	6		
PurchaseOrderTyp Click to filter>	pe		□ I Crick to filter>		
(3) orderDate	[01] date	Move 🕶	() orderDate)1] date	1
e custID		Move , Dre cre	customerID [1	L.,1] string	1 -
	🥵 XPath Expression B	uilder	×	.1] OrderAddress	1
🕀 🖻 shipTo	Build an XPath Expr	ession	11	.1] OrderAddress	1
🕀 🖻 billTo		on below with the help of content assist.	Use "Insert Simple	.1] Items	1
🕀 🖲 items	XPath" if you need more	guidance.	//		10
	Insert Simple XPath				•
뿝 Task Flows 🙆 Build Act	<type an="" expression<="" td="" xpath=""><td>on here. Content assist available (Ctrl+s</td><td>bacé)></td><td></td><td>▽ ¤ E</td></type>	on here. Content assist available (Ctrl+s	bacé)>		▽ ¤ E
🕾 Transform - If			*		
Cardinality The in	Content assist available (C	[trl+space)		d to the input element.	
Condition			OK Cancel	8	Edit
Order					
Documentation					

_____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.

🚯 Simple XPath Expression Builder	x
Build a simple XPath Expression Select a field and optionally add filters and conditions.	55
Expression:	
Select a field:	
<type filter="" to=""></type>	•
Data Types	
Add an optional filter	

- ____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).

😤 *OrderMapProcess	🔥 *MapO	rder 🛛			
🏝 MapOrder					
⇒ <click filter="" to=""></click>	,		⇒ <click filter="" to=""></click>		
(a) orderDate	[01]	date	Move - (a) orderDate	[01]	date
e custID	[11]	strina	Move - e customerID	[11]	string
		1] Address		[01]	OrderAddress
	[0, 1]			[01]	OrderAddress
	[01]		Else if Else i	[01]	Items
	[01]	Address			

- 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.

CUSCID	[11] string	E e shipToAddress [01] OrderAddress
€ € shipTo	[01] Address	B V If ■ e billToAddress [01] OrderAddress Else if ■ e items [01] Items
🕀 🖻 billTo	[01] Address	
🗉 🖻 items	[01] Items	Local map - 👻
Task Flows R Task Flows	uild Activities 🔲 Properties 🕅	💽 Problems) 👸 Server Logs) 👯 Servers) 🐻 Asset Repositories) 🔂 Test Map) 🛛 🖓 🖓
🕾 Transform - E		
		ainst the condition. If the condition evaluates to true, the transform is applied to the input element.

- _____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.

*OrderMapProcess	🖳 *MapOrder 🛛			E
KapOrder				
•MapOrder	10 10 X 10 1	r 🔒 🗳 🛱 ன 🖾	13	
☐ 2 PurchaseOrderTy	pe		□ 🔛 OrderType 券 <click filter="" to=""></click>	
(a) orderDate	[0.,1] date	Move -		01] date
e custID	[01] date	Move , C.a. ore		1,.1] string
COSCID	🪯 XPath Expression E	Builder	×	.1] OrderAddress
🕀 🖻 shipTo	Build an XPath Exp	ression	11	.1] OrderAddress
🕀 🖻 billTo		ion below with the help of content assist.	Use "Insert Simple	.1] Items
🕀 🖻 items	XPath" if you need more	e guidance.	/ /	
4	Insert Simple XPath			
Task Flows 🙆 Build Ac	<type an="" express<br="" xpath="">t</type>	ion here. Content assist available (Ctrl+s	pacé)>	~ □ [
🚭 Transform - If			*	
Cardinality The in	Content assist available ((Ctrl+space)		d to the input element.
Condition	0		OK Cancel	Edit
Order				
Documentation				-

_____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.

🚯 Simple XPath Expre	ession Builder				×
Build a simple XPat Select a field and option	•	onditions.			55
Expression:					
\$shipTo1/@country =	= 'CANADA'				
Select a field:					
<type filter="" to=""></type>					•
 State State State State State Street Street City : str 	e : string string ring				
Array Fields	Where	Operator	Value	and/or (optional)	Delete
Example					
Add an optional co	ondition				
Operator = Example	CANADA]			Delete
0				ок	Cancel

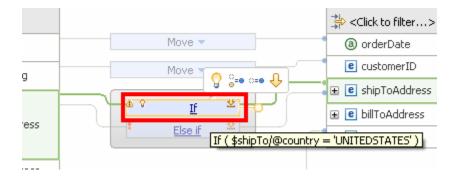
____h. Notice that the expression is being built at the top of the builder (in bold text). Click OK.

____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.**

🖶 XPath Expression Builder		×
Build an XPath Expression		1
Type an XPath expression below with the help of content ass XPath" if you need more guidance.	ist. Use "Insert Simple	\sum
Insert Simple XPath		
[©] \$shipTo1/@country = 'CANADA'		_
		v
Content assist available (Ctrl+space)		
0	ок	Cancel

- ____j. Save the changes. From the menu select **File** \rightarrow **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.

😤 *OrderMapProcess	×1v	1apOrder 🕅									
🌯 MapOrder ゝ 🖳 st	nipToAdd	ress			_	_					
•MapOrder 🧳	a 100	₽ X I	>ı ţı⊅	8	1	\$) (8	0- 0-				
						Map sou	irce to ta	arget base	d on name and types	2	
🖃 泡 shipTo	[01]	Address				_			🖃 🖳 shipToAddress	[01]	OrderAddress
⇒ <click filter="" to=""></click>									♣ <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN							a country	[01]	NMTOKEN
e firstName	[01]	string							e name	[01]	string
e lastName	[01]	string							e street	[01]	string
e street	[01]	string							e city	[01]	string
e city	[01]	string							e state	[01]	string
									e zip	[01]	string

____ 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.

😤 *OrderMapProcess	🔁 *M	1apOrder 🔀								
🍋 MapOrder 🔰 🖳 shipToAddress										
•MapOrder 🛛 🖓	- 1 0-	₽× ↓	¢ı ↓ r⊳	R. []	₿ 👪	0- 0-				
								1	5	
🖃 🙋 shipTo	[01]	Address						🖃 💽 shipToAddress	[01]	OrderAddress
⇔ <click filter="" to=""></click>								\Rightarrow <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN			Move 🔻		•	(a) country	[01]	NMTOKEN
e firstName	[01]	string	-0					e name	[01]	string
e lastName	[01]	string						e street	[01]	string
e street	[01]	string	Add Co	onnection	Move 🔻			e city	[01]	string
		-			Move 🔻			e state	[01]	string
e city	[01]	string						e zip	[01]	string

占 MapOrder 👌 💽 s	shipToAddi	ress							
•MapOrder	p1 40	₽ X ↓	iþi ∫ti⊅	R. []	₿ 🖬	0- 0-			
							5	2	
🖃 🔁 shipTo	[01]	Address					🖃 💽 shipToAddress	[01]	OrderAddress
sclick to filter>	,						Click to filter>		
() and ()	[01]				Move 🔻		@ couptry	[0 1]	NMTOKEN
e firstName	[01]	chrina			Move 🔻		e name	[01]	string
		-					E shoul	[01]	Juling
e lastName	[01]	string	_				e city	[01]	strina
e street	[01]	string			Move 🔻		e state	[01]	-
	[01]				Move 🔻		e zip		

_____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

😤 *OrderMapProcess	× *	1apOrder 🛛									- 0	
🖳 MapOrder 👌 🕄 shipToAddress												
•MapOrder 🧳	a 16	₽ × ↓	¢ı ≬a ¢ (R 🗳	₿ 👪	0-	The "Move" inputs.	mapping was updated	to a "Cor	ncat" mapping to	accommodate multiple	
	[0, 4]	Address							<u> </u>	0.1.1.1.1		
□ 1 shipTo ⇒ <click filter="" to=""></click>		Address						■ ShipToAddress ⇒ <click filter="" to=""></click>		OrderAddress		
(a) country	[01]	NMTOKEN	[Move 🔻			(a) country	[01]	NMTOKEN		
e firstName	[01]	string			Concat 👻			e name	[01]	string		
		· · ·		`				· •	e street	[01]	string	
e lastName	[01]	string	_					e city	[01]	string		
e street	[01]	string	l		Move 🗸			e state	[01]	string	1	
e city	[01]	string			Move 🔻			e zip	[01]	string		
			_									

____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.

🗄 MapOrder 👌 😨 shipToAddress								
MapOrder	91 ID	₽× .	ida inda	R. 🗳 🛱 👪	0-			
						2	2	
🖃 🔁 shipTo	[01]	Address				🖃 🖳 shipToAddress	[01]	OrderAddress
🔆 <click filter="" to=""></click>	,					⇒ <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN		Move 🔻		a country	[01]	NMTOKEN
e firstName	[01]	stripa		Concat 👻		e name	[01]	string
		_	ſ			e street	[01]	string
e lastName	[01]	string				e city	[01]	string
e street	[01]	string		Move 🔻		e state	[01]	string
e city	[01]	-h		Move 🔻		e zip	[01]	string

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

Part 4: Cut, Copy, and Paste transforms

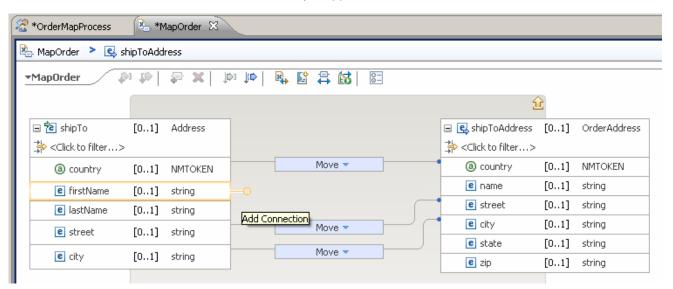
In this part of the lab, you will finish connecting the map inside the Else-If transform. To help you do this, you will copy the Concat transform inside the If map and paste it into the Else-if map. Version 7 XML Mapping Editor has the ability to cut, copy, and paste transforms. This option speeds development 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.
 - 😤 *OrderMapProcess 🏊 MapOrder 🛛 🗙 🍋 MapOrder MapOrder 🔊 💭 jþi jþ × L¢. 🛱 👪 🛛 8-× 🖃 🛜 PurchaseOrderType 🖃 🔣 OrderType 🔆 <Click to filter...> 🔆 <Click to filter...> Move 🔻 a) orderDate [0..1]date (a) orderDate [0..1] date e customerID [1..1] string Move e custID [1..1] string 표 🖻 shipToAddress [0..1] OrderAddress °=• ः• ⊕ TE 🗄 💽 billToAddress OrderAddress [0..1]표 🖻 shipTo [0..1] Address Else if [0..1] 표 💼 items Items 표 🖻 billTo [0..1] Address Local map 👻 표 💽 items [0..1] Items
 - ____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.

🕈 *OrderMapProcess 🥄 🍇 *MapOrder 🛛											
🏝 MapOrder ゝ 🕵 sl	hipToAdd	lress			_						
<u>▼MapOrder</u>	N III	₽× ↓	⊳ı ţı⊳	×	E	t z	0-				
					м	ap sou	rce to targ	et base	d on name and types	2	
🖃 🔁 shipTo	[01]	Address							🖃 🖳 shipToAddress	[01]	OrderAddress
⇒ <click filter="" to=""></click>									♣ <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN							a country	[01]	NMTOKEN
e firstName	[01]	string							e name	[01]	string
e lastName	[01]	string							e street	[01]	string
e street	[01]	string							e city	[01]	string
e city	[01]	string							e state	[01]	string
									e zip	[01]	string
									-		

____ 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.

*OrderMapProcess	*OrderMapProcess 🖳 *MapOrder 없								
🏝 MapOrder 🕨 🖳 sh	ipToAdd	ress							
•MapOrder 🧳	- 1	₽×↓	DI ID	R. E	a 🛱	0-			
							5	8	
🖃 🔁 shipTo	[01]	Address]				🖃 💽 shipToAddress	[01]	OrderAddress
⇒ <click filter="" to=""></click>							⇒ <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN			Move 🔻		a country	[01]	NMTOKEN
e firstName	[01]	string	-				e name	[01]	string
e lastName	[01]	string	-				e street	[01]	string
e street	[01]	string	-		Move 🔻		e city	[01]	string
		-			Move 🔻		e state	[01]	string
e city	[01]	string			1.010		e zip	[01]	string

- Move
 Image: Second second
- ____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.

😤 *OrderMapProcess	×.	1apOrder 🔀							
🏝 MapOrder 🜔 🖳 s	hipToAdd:	ress							
•MapOrder	4 1 , 16	₽ X .	joi to	R. []	₿	0-			
							2	2	
🖃 泡 shipTo	[01]	Address					🖃 🛃 shipToAddress	[01]	OrderAddress
🔆 <click filter="" to=""></click>							⇔ <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN			Move 🔻		 @ country 	[01]	NMTOKEN
e firstName	[01]	string			Concat 👻		e name	[0_1]	string
		-				- 💛 Undo		.1]	string
e lastName	[01]	string				Redo		.1]	string
e street	[01]	string			Move 🔻	Revert		.1]	string
e city	[01]	string			Move 🔻	of Cut	Ctrl+X	.1]	string
		-				📔 Copy	Ctrl+C		
						Paste	Ctrl+V		
						_			

____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.

😓 MapOrder 🚬 🖳 shipToAddress								
MapOrder	1 (l) (l)	₽ X .	¢1 ↓¢	ୠ 😰 🛱 👪 🖂				
						2	2	
🖃 泡 shipTo	[01]	Address				🖃 💽 shipToAddress	[01]	OrderAddress
🔆 <click filter="" to=""></click>	,					🔆 <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN	-	Move 🔻]	(a) country	[01]	NMTOKEN
Cinebalance	FO 11	-hui	_	Concat 👻	<u>⊢</u>	e name	[01]	string
e firstName	[01]	string		1	' <u> </u>	e street	[01]	string
e lastName	[01]	string				e city	[01]	string
e street	[01]	string		Move 🔻		e state	[01]	string
e city	[01]	string		Move 🔻	\vdash	e zip	[01]	string

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

😤 *OrderMapProcess	🔥 MapOre	der 🛿						
🏝 MapOrder								
•MapOrder 🔊	P =	X []]	· 🗳 🤇	🛱 👪 😂				
🖃 ն PurchaseOrderTy	/pe					🖃 🛃 OrderType		
⇒ <click filter="" to=""></click>						🔆 <click filter="" to=""></click>		
(8) orderDate	[01]	date		Move 🔻		(a) orderDate	[01]	date
e custID	[11]	string		Move 🔻		e customerID	[11]	string
		-		TF 8=0 0=0 1		🗄 🖻 shipToAddress	[01]	OrderAddress
	[0 1]	Address		1	J	🗉 🖻 billToAddress	[01]	OrderAddress
a e snpro	[01]	Hadress	Ľ	<u>Else if</u>		🗄 🖻 items	[01]	Items
± € billTo	[01]	Address						
	[01]	Items		Local 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**.

😤 *OrderMapProcess	×. *	MapOrder	×							
🏝 MapOrder ゝ 😫 st	hipToAd	dress								
-MapOrder 👘	₽	₽ ×	jd>ı †d⊅	R. E	° 📮 🕯	5				
									Ê	
🖃 泡 shipTo 🛛 🛛	[01]	Address						🖃 💽 shipToAddre	ss [01]	OrderAddress
🛟 <click filter="" to=""></click>								🔆 <click filter.<="" td="" to=""><td>></td><td></td></click>	>	
(a) country	[01]	NMTOKEN			Move	•		(a) country	[01]	NMTOKEN
e firstName [[01]	string					0	e name	Undo	
e lastName [[01]	string						e street	Redo	
e street	01]	string			Move	•		e city	Revert	
	 [0 1]	string		-	Move	-		e state	Cut	Ctrl+
e city [[01]	sunny						e zip	Gipy	Ctrl+
									💼 Paste	Ctrl+'

_____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.

🧏 *OrderMapProcess 🚯 *MapOrder ⊠								
🏝 MapOrder ゝ 🖳 shipTo	Address							
•MapOrder 🛛 🔑	₽×↓	⊳ı ţı¢	🗣 🗳 🛱 🚺 🖂					
					2	2		
🖃 泡 shipTo 🛛 [01]	Address				🖃 💽 shipToAddress	[01]	OrderAddress	
Click to filter>					⇔ <click filter="" to=""></click>			
(a) country [01]	NMTOKEN	-0	Move 💌		a country	[01]	NMTOKEN	
e firstName [01]	string		Concat 🗸		e name	[01]	string	
	-				e street	[01]	string	
e lastName [01]	string				e city	[01]	string	
e street [01]	string		Move 🔻		e state	[01]	string	
e city [01]	string	1	Move 🔻		e zip	[01]	string	

____ 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.

	G country	fourth in contrast		-
	e firstName	e [01] string	-O Concat -	e nai
	e lastName	e [01] string		e str
	e street	[01] string	Move 🔻	e city
			Move 🔻	e sta
	e city	[01] string	- Hove -	e zip
88	Task Flows 🙆 B	uild Activities 🔲 Properties	इ 🔀 🔣 Problems) 👸 Server Logs) 🖧 Server:	s 🕼 Asse
6	Transform -	Concat		
Ge	eneral	Example: <firstname><la< th=""><th>stName></th><th></th></la<></firstname>	stName>	
Ca	rdinality	Prefix:		
Co	ndition	• User defined value:	O Space character	
	der	Default delimiter:		
Do	cumentation	$oldsymbol{\Theta}$ User defined value:	O Space character	
		Input	Delimiter	
		firstName		
		lastName		
			Restore default de	elimiter
		Suffix:		
		$oldsymbol{\Theta}$ User defined value:	O Space character	

- ____I. 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** \rightarrow **Save** (or press **Ctrl** + **S**)

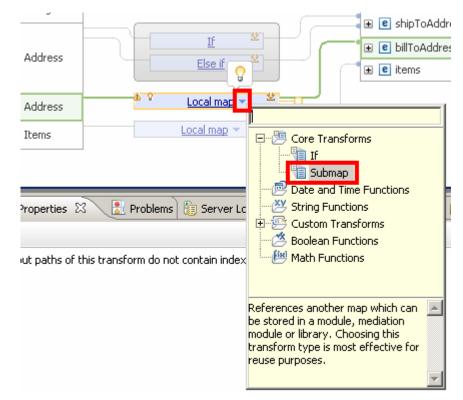
Part 5: Refactor to a submap

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.

🔏 *OrderMapProcess 🛛 🔌 MapO	rder 🛛					
🏝 MapOrder						
MapOrder 🔊 🌮 🍃	- X 1	🏴 🖳 😭 🛱	8	 		
🖃 🗽 PurchaseOrderType				🖃 🔛 OrderType		
Click to filter>				\Rightarrow <click filter="" to=""></click>		
(a) orderDate [01]	date		Move 🔻	 (a) orderDate	[01]	date
e custID [11]	string	·	Move 🔻	 customerID	[11]	string
	-		TF Y	🗄 🖻 shipToAddress	[01]	OrderAddress
	Address		<u>11</u>	🗉 🖻 billToAddress	[01]	OrderAddress
	11001000		Else if 🛛 💆	🗄 🖻 items	[01]	Items
• e bilTo [01]	Address	-0	Local man 🗶 🖄			
	Items	Add Connection	Local map 👻 🎽			

_____ 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**.

	 ● shipTo ● billTo 		[01]	Address			Else if Submap 🔻	Y			e items	[01]
			[01]	Address				芝				
	표 🖻 items		[01]	Items			Local map					
88	Task Flows	Build Activ	vities 🔲	Properties	×	Problem:	s 👸 Server	Logs 👯	Servers	🔓 Assel	t Repositories	🔂 Test Map
6	Transform -	Subma	р									
Ge	neral	File:	MapOrd	er.map					Bro	wse	New	
Ca	rdinality	Map:	2						•			
Co	ndition											
Or	der											
Do	cumentation											

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.

🔠 Task Flows 🗟 B	Build Activities 🔲 Properties 🛛 🖹 Problems 👸 Server Logs 👫 Servers 陆 Asset Repositories
🗟 Transform -	Submap
General	File: MapOrder.map Browse New
Cardinality	Map: 🙎
Condition	
Order	
Documentation	
	1

____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**.

🚯 New XML Map)	_ 🗆 🗙
Create an XMI XML maps are use	L Map ed to transform data between source and target business objects.	X
Module or library:		
Namespace:	http://MappingExercise/AddressToOrderAddress 🔽 Default	
Folder:	Browse	
Name:	AddressToOrderAddress	
0	< Back Next > Finish	Cancel

____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**.

占 AddressToOrderAd	ldress								
•AddressToOrder#	Address	I II	₽ ×	jdı îd	8	8 📳 🔂	0-		
🖃 🔁 Address						Map sourc	e to target based on na	me and ty	/pes
🔆 <click filter<="" td="" to=""><td>.></td><td></td><td></td><td></td><td></td><td></td><td>🔆 <click filter="" to=""></click></td><td></td><td></td></click>	.>						🔆 <click filter="" to=""></click>		
a country	[01]	NMTOKEN					(a) country	[01]	NMTOKE
e firstName	[01]	string					e name	[01]	string
e lastName	[01]	string					e street	[01]	string
e street	[01]	string					e city	[01]	string
e city	[01]	string					e state	[01]	string
							e zip	[01]	string

- ____ 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)

😤 *OrderMapProcess	🔁 *M	lapOrder	🕭 Address	ToOrderAddress 🕱				
AddressToOrderAd	ldress							
AddressToOrderA	ddress	P1 🕼	₽ ×		₿ 👪	0-0-		
🖃 🗽 Address						🖃 🜆 OrderAddress		
🛟 <click filter<="" td="" to=""><td>.></td><td></td><td></td><td></td><td></td><td>\Rightarrow <click filter="" to=""></click></td><td></td><td></td></click>	.>					\Rightarrow <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN	[Move 🔻	-	a country	[01]	NMTOKEN
e firstName	[01]	string		Move 🔻	•	e name	[01]	string
e lastName	[01]	string	-			e street	[01]	string
		-		Move 🔻		e city	[01]	string
e street	[01]	string		11046 +		e state	[01]	string
e city	[01]	string		Move 🔻		e zip	[01]	string

____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**.

😵 *OrderMapProcess 🛛 🕾 *MapOrder 🗙	AddressToOrderAddress	
🖳 MapOrder		
-MapOrder 🔊 🌮 🗁 🗶 🕬	[10] 🗣 😫 🚍 👪 🛛 📰	
🖃 ն PurchaseOrderType		🖃 💽 OrderType
⇒ <click filter="" to=""></click>		⇒ <click filter="" to=""></click>
(a) orderDate [01] date	Move 🔻	(a) orderDate
custID [11] string	Move 🔻	customerID
		표 🖻 shipToAddress
🗄 🖻 🕂 Save Changes		€ billToAddress
🙂 🖻 🕯 🔂 Save Changes	×	🗄 🖻 items
 The editor has detected a chan needs to reload. Any changes to save your changes before the r € i 	ge in the input or output of this map, and that you made here will be lost. Do you want to map is reloaded?	
	Yes No	

									. (1	+ e	shipToAc	Idress
🛨 🖻 shipTo		[01]	Address			<u>If</u> Else	if	¥ ¥	F	ſ			billToAdo	
🛨 🖻 billTo		[01]	Address			Subma	p 🔻	⊻]					
🕀 🖻 items		[01]	Items			Local ma	ар 🔻	堂						
Task Flows	Build Activ	vities 🔲	Properties	ଞ୍ଚ 🖹 Pi	oblems	👘 Ser	ver Lo	ogs 위	لا Serv	ers	💪 Ass	et Rej	positorie	s) 🔂 Te
器 Task Flows 心 る Transform -			Properties	ध्र 💽 Pi	roblems		ver Lo	ogs 🖣	성 Serv	ers	🔓 Ass	iet Rej	positorie	s) 🔂 Te
		ip	Properties sToOrderAdd		roblems	i Ser	ver Lo	ogs 👌 🖨	상 Serv		🔓 Ass	1	positorie ew	s) 🔂 Te
🕾 Transform -	Subma	Address		dress.map	roblems	i Ser	ver Lo	ogs 여	∜ Serv			1		s) 🔂 Te
General	Subma	Address	sToOrderAdd	dress.map	roblems	Ser	ver Lo	ogs) 바	Serv			1		s) 🔂 Te
General Cardinality	Subma	Address	sToOrderAdd	dress.map	roblems	i Ser	ver Lo	ogs) 바	∜ Serv			1		s) 🔂 Te
General Cardinality Condition	Subma	Address	sToOrderAdd	dress.map	roblems		ver Lo	ogs) 위	Serv			1		s) 🚺 Te

____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 \rightarrow 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**.

			-			<u>v</u> -		€ :	hipToAddress;
		[01]	Address		<u>11</u>	× - * -		± e t	billToAddress
					<u>LISE II</u>			∎ <u>e</u> i	tems
🛨 🖻 billTo		[01]	Address		<u>Submap</u> 🔻	Ç	Undo Change su	ubmap	
🗉 🖻 items		[01]	Items		Local map 🔻		Redo		
]			Revert		
						ot	Cut		Ctrl+X
법 Task Flows (중)	n, dal Alaca	au	Duna subia a	3 💦 Probl	ems) 🛅 Server Log	P	Сору		Ctrl+C
	שטוום אכנוי		Properties 2		ems 🛛 🔠 Server Log		Paste		Ctrl+∀
🕾 Transform -	Subma	р				"	- Create Connect	ion.	
General	File:	Address	sToOrderAddr	ess.map			Edit		
Cardinality	Map:	Addres	sToOrderAdd	ress			Expand All		
Condition						~			
Order							Delete		
Documentation							Sort Transforms	;	
							Change transfo	rm	
						×	Associate XML		Ctrl+Shift+X
						Ľ	Generate XSLT		Ctrl+Shift+G
							Match Mapping		Ctrl+Shift+M
						1	Refactor from S	iubmap	Ctrl+Shift+E

____b. The submap is now changed back to a local map.

		te the ship loAddress [U1] OrderAddress
	[01] Address	· · · · · · · · · · · · · · · · · · ·
	[0.1] Mudicas	Else if
🛨 🖻 billTo	[01] Address	
	[01] Items	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.

😵 *OrderMapProcess	№ *M	lapOrder 🕅	🚴 Ad	dressTo	OrderAddress					
🏝 MapOrder										
MapOrder 🧳	1 (₽× ↓	þ1 ļu⊳	B	ŝ 🛱 🙀	0- 0-				
								2	2	
🖃 泡 billTo		Address]					🖃 💽 billToAddress		OrderAddress
🔆 <click filter="" to=""></click>								\Rightarrow <click filter="" to=""></click>		
(a) country	[01]	NMTOKEN		-	Move 🔻		•	a country	[01]	NMTOKEN
e firstName	[01]	string		-	Move 🔻			e name	[01]	string
		-						e street	[01]	string
e lastName	[01]	string			Marra		ך נ	e city	[01]	string
e street	[01]	string			Move 🔻			e state	[01]	string
e city	[01]	string			Move 🔻			e zip	[01]	string

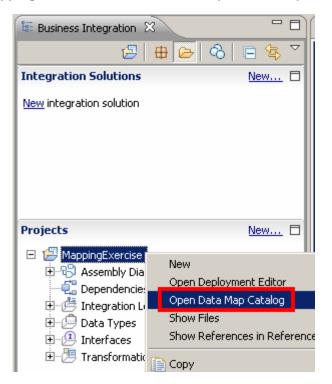
____ d. Save the changes. From the menu select File \rightarrow Save (or press Ctrl + S)

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.

*OrderMapProcess 🛛 🚱 MapO	rder 🛛 🔼 4	AddressToOrderAddress	🔚 Data Map Catalo	g - MappingExercise	×	- E
Data Map Catalog 🛛 📑 👔	⇒ x <i>A</i>	↔ 冊 惣			Scop	e: MappingExercise
Showing 2 XML Maps						
Name	↓ [±] Type [±] ↓ [±]	Inputs ±	t Outputs 44	Location	[±] [±] Documentation [±] [±]	
AddressToOrderAddress	XML Map	📮 Address	🗐 OrderAddress	/MappingExercise		
🍋 MapOrder	XML Map	📋 PurchaseOrderType	📋 OrderType	/MappingExercise		

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

🔋 *OrderMapProcess 👘 🔼 MapC	rder 🔼	AddressToOrderAddress	📑 Data Map Catalo	g - MappingExercise	×	- 6
Data Map Catalog 🛛 📑 👔	⇒ × <i>A</i>	□ ↔ 冊 < 控			Scop	e: MappingExercise
Showing 2 XML Maps						
Name	[±] Type [±] + [±]	Inputs ±	[±] Outputs [±] ₄ [±]	Location	[±] [±] Documentation [±] [±]	
AddressToOrderAddress	XML Map	🗐 Address	CrderAddress	/MappingExercise		
🏝 MapOrder	XML Map	📋 PurchaseOrderType	📋 OrderType	/MappingExercise		

____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.

😤 *OrderMapProcess 👘 🔼 Ma	pOrder	R.	AddressToOrderAddress		🕞 Data Map Cata	log	- MappingExercise	x			-
Data Map Catalog 🛛 📑	⊳ x	1	○ 冊 惣						Sc	оре	: MappingExercise 🔻
Showing 2 XML Maps											
Name	±± Тур	e 11	Inputs	±_±	Outputs *	t_t	Location [±] ,	ļ±	Documentation	±_±	
AddressToOrderAddres		. Map	Address	<u> </u>	OrderAddress	_	/MappingExercise	<u> </u>	Documentation	<u> </u>	
🖳 MapOrder	XML	Мар	DurchaseOrderType		CrderType		/MappingExercise				

____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.

(😤 *OrderMapProcess 👘 🔼 Map	Order	R. (AddressToOrderAddress		🚰 Data Map Cat	aloç:	g - MappingExercise	2	3		E
	Data Map Catalog	6 🗙	ß	🕶 🖶 惣						Scop	e: MappingExercise	Ī
l	Name	±± Type	± ₄ ±	Inputs	±4;∓	Outputs	± ₄ ±	Location	±4∓	Documentation ±±		1
L	🍇 AddressToOrderAddress	XML N	1ap	🚊 Address		🚊 OrderAddress		/MappingExercise				
	🍋 MapOrder	XML N	1ap	📋 PurchaseOrderType		📋 OrderType		/MappingExercise				
l												

- 😤 *OrderMapProcess 🍋 MapOrder AddressToOrderAddress 🚡 Data Data Map Catalog 뉗 đ <.> C-Ś Showing 2 XML Maps X Find ±₄± T Name AddressToOrderAddress Find: MapOder Ŧ Xľ Ac 🏝 MapOrder T১ Xľ Direction Options Forward Wrap search Case sensitive Backward Find Close
- ____f. You can also find a data map using the toolbar Find icon.

____g. You can show content tagging and namespace information

 *OrderMapProcess	🔥 MapOrder	AddressToO	rderAddress	🔚 Data Map Catalog) - MappingExercise	×
Data Map Catalog	a 🖻 🗙	A 🛛	** 2			
Name	±± Type	±₊± Inputs	±4‡	Outputs ±±	Location	±₊± Documer
🐁 (http://Mapping	gExercise/Add XML N	4ap 🛛 📋 {http:/	/www.ibm.co	📋 {http://www.i	/MappingExercise	
🐁 (http://Mapping	gExercise/Map XML N	4ap 🛛 📋 {http:/	/www.ibm.co	📋 {http://www.i	/MappingExercise	

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.

😵 *OrderMapProcess	🔥 MapOrder	AddressToOrderAdd	dress 🛛 🔚 Data Map	Catalog 🛿			
Data Map Catalog Showing 1 XML Map	📑 🗁 🗙	<i>∦</i> ⇔ ⊕ ½				Scope: <a>Workspace <a>Workspace MappingExercise	
Name <add> 🔺</add>	📿 🖽 Type	[±] ± [±] Inputs	4 [±] Outputs	4± Location	±± Documental	tion <u>**</u>	
AddressToOr	derAddress XML M	1ap 📋 Address	🛱 OrderAddr	ess /MappingExei	rcise		

____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