



IBM Integration Bus

Pattern Authoring Lab 3

Setting parser properties with XPath expressions

June, 2013

Hands-on lab built at product code level version 9.0

1. Lab Objectives

In this lab, you will see how to configure the parser properties on an MQInput node. Configuring the message domain, message model, message and physical format properties on input nodes is a very common requirement in pattern authoring.

This lab uses a simple message flow with an MQ Input node. The lab will develop the pattern built in the first and second labs. The MQ Input node requires the specification of parser properties (even if defaulted). If the XMLNSC domain is selected, the message model, message and physical format parameters are not used. If the MRM domain is selected, the other parser parameters are required. This pattern enables these parameters to be specified according to selected domain.

The starting point for this demonstration is the same message flow and pattern that were created in labs 1 and 2.

This approach can be extended to many similar functions with the Integration Toolkit.

Extending the pattern to specify parser properties

1. Start this lab with the Pattern Authoring editor for the pattern "MyPattern", as used in the previous lab session.

If you have still got the second open instance of the Integration Toolkit, close it now, and use the primary instance.

Select the message flow Transform.msgflow (you may need to open it from the Transform project).

📧 Transform.msgflow 🛛 🗎	🔡 MyPattern.patt	tern	- E
👌 🔅 Palette			
Selection			
Note			
Umis JMS			
HTTP	MQ Inp	put) Transform MQ Output	
Web Services			
Let SCA			
WebSphere Adapters	1		
C Routing			
🔁 Transformation 🛛 🗠			
Construction			
CO Database			
🥪 File 🗸 🗸	1	Log queue output	
Graph User Defined Propertie	es		
🔲 Properties 🕺 🚼 Pro	oblems 🔠 Deployme	ent Log) 🖳 Console) 🗢 🌣	- E
🛃 MQ Input Node Pro	perties - MQ Inp	put	
Description			
Basic	Message domain	<select a="" blob="" default="" domain="" in="" is="" message.="" no="" present="" the="" to="" use="" when=""></select>	•
Input Message Parsing	Meccane cet		-
Parser Options	mossage set		
Advanced	Message type		_
Validation	Message format		•
Security			
Instances			

This screen shows the message parsing properties of the MQ Input node. The default message domain is BLOB. This pattern will allow the pattern user to choose between XMLNSC and MRM. Other options will be removed from the selection list.

2. Right-click the MQ Input node, and select Patterns -> Select Target Properties (or you can click on the icon directly).

Expand the MQ Input node, expand Input Message Parsing.

Select all the properties in the parsing group (or select the parsing group itself).

Close the Target Properties pop-up.



Save the updated message flow, and close the flow editor.

3. In the pattern editor, select the "Source Files" tab. See the new Target Properties have been added (right pane).

**Wynattern, paitern Image: Transform.magflow Source Files Select the source files to include in your pattern. You can also view the to get p To select target properties, select Your source file and click "Select Target Proper Then right-click a node or the canvas and select "Patterns > Select Target Properties"	rsperties available in the source files. Ittles", or double-oldk your source file. arbes",
Select Source Files	Target Properties
Image: Strate State Image: State Image	Head text Head text
Source Files Pattern Configuration Categories Create Fattern	

Switch to the Pattern Configuration tab. 4.

> The parser properties are initially not in the correct logical order. Use the Up and Down buttons (highlighted) to correct the order. (For example, highlight "Message Set", and click the Up button).

Alternatively, you can drag and drop the Parameter names within the list.

Change the order to:

- 1. Message Domain
- Message Model
 Message
- 4. Physical format

Transform.msgflow	
Using pattern parameters and your own logic, you can extend your pattern with <u>Dava and PHP code</u> .	
Groups and Parameters	
Application Queue Information	💽 Add Group
E' = Queue name (queuename)	
Sets property: Transform.indgl.Transform.indgl_input.queuewame Sets property: Transform.indgl.Transform.indgl_input.queuewame	Add Parameter
	🕞 Edit
E E MQ_Input	
i li → 😭 Message domain (pp1)	Delete
E Sets property: Transform.mqsi.Transform.MQ_Input.messageDomainProperty	
E Physical forma (pp2)	Enumerated Types
P → Message (no)	Tables
Line Sets prometry: Transform.mgi.Transform.MQ. Input.messaneTyneProperty	
□ ⇔ Message model (pp4)	2 3
Sets property: Transform.mgsi.Transform.MQ_Input.messageSetProperty	
	📻 Expand All
Source Files Pattern Configuration Categories Create Pattern	

5. Finally, change the name of the parser properties group for the MQ Input node. Double-click the MQ_Input group, and change the name as shown.

Click	OK
Olicit	UIN.

dit Group: MQ_Input			
nfigure group			
Configure the pattern parame Configure an XPath expressio	eter group and how it is o In that controls when this	displayed to pattern users. s group is enabled in the Pattern Instance editor.	
sic Enable			
Group Display	MO. Input Node	a Darray Dronayties	
Display frame:	I wo Tubar wage	raiser roperues	
Description:	Pattern Parame	ters	
Group Options			
Generate help documer	itation	Select this option to create help information for this group in the pattern documentation.	
 Display parameters in a 	group box	Select this option to display this group, and any parameters it contains, with a surrounding box.	

6. After making these changes, you should have something like this:

🕫 Transform.msgflow 🔀 *MyPattern.pattern 🛛	
Configure your groups and pattern parameters and associate the pattern parameters with their target properties.	
Using pattern parameters and your own logic, you can extend your pattern with <u>Java and PHP code</u> .	
Groups and Parameters	
P-II Application Quarter Deformation	🔚 Add Course
	Kuu Group
E Course name (deconvante)	⇔
Sets property: transformings, transforming, fransforming, input, duedename	E Add Parameter
E Queue sumx (queuesumx)	Edit
Here Mc Input Node Parser Properties	
E = Message domain (pp1)	Delete
Sets property: Transform.mqsi.Transform.MQ_Input.messageDomainProperty	
E- ≧ Message model (pp4)	Enumerated Types
Sets property: Transform.mqsi.Transform.MQ_Input.messageSetProperty	
Ē─ 🚆 Message (pp3)	Tables
⊡− 😭 Physical format (pp2)	2 3
Sets property: Transform.mqsi.Transform.MQ_Input.messageFormatProperty	
	∓ Expand All
	📄 Collapse All
Source Files Pattern Configuration Categories Create Pattern	

7. The Message Domain is an interesting target property, because it has a pre-defined list of permissible values. This is called an enumerated type in pattern authoring. We will configure constraints on this property now, to reflect these values.

Click Message Domain, and click Enumerated Types. This will open this window.

This dialogue allows you to look at and configure enumerated types. When the message domain target property was added to this pattern, the Pattern Authoring editor automatically created an enumerated type for it. The permissible values for the enumerated type are shown here. Each entry has a display name, and a value which the target property can be configured with (such as MRM and XMLNSC).

e value is the content that is stored in the target property and can be alphan	ance of this nattern		
Enumerabed type: Message domain	ince of ville pacetini interio or '_ only.		_ _
	Add	Remove Rena	me Duplicate
)isplay Name Value			
)FDL : For binary or text messages with a Data Format Description Lan DFDL			odd
MLNSC : For XML messages (namespace aware, validation, low me XMLNS	c		Ann
)ataObject : For data from WebSphere Adapters, CORBA and Data DataC	oject		_
ISON : For JavaScript Object Notation messages JSON			Remove
SLOB : For messages with an unspecified format BLOB			
4IME : For MIME wrapped data including multipart MIME			
1RM : For binary or text messages that are modeled in a message set [MRM]			
MSMap : For JMS MapMessage messages (XML) JMSMa	p		
MSStream : For JMS StreamMessage messages (XML) JMSSt	eam		
(MLNS : For XML messages (namespace aware) XMLNS			
			Reset Values
			KC3CC VOIDC3

8. We will remove all the entries in this list except for XMLNSC and MRM. If you make a mistake, you can reset this list by clicking on the Reset Values button below. You can also see at the bottom of this dialog, that the message domain pattern parameter is using this enumerated type.

Select the unwanted lines from the list, and click Remove.

Click OK to return to the Pattern Configuration window.

			Add	Remove	Rename	Duplicate
isolay Name		Yalue				
nspray nyante MLNSC : For XML messages (n 1904 : Fan bia ann an barb	amespace aware, validation, low me	XMLNSC				Add
RM : For binary or text messa	iges that are modeled in a message set					Remove
						Reset Values
i cannot remove an enumerat	ed type if it is being used by a paramete	er or is defined by a ta	raet property,			10000 10000

9. At this point, it would be sensible to change the generated pattern parameter ID to something more useful. For the Message Domain, we will change it to MessageDomain.

To do this, highlight the Message Domain property, and click Edit. Change the Parameter ID field, and click OK.

을 Edit Parameter: Message domain	x
Configure pattern parameter	
Configure the pattern parameter and how it is displayed to pattern users.	
Basic Editor Transform Enable Parameter Display	
Parameter Options	
Hide the parameter Select this option to hide the parameter and to use an XPath expression to set the value of the parameter when a pattern instance is created.	
Image: Mandatory parameter Select this option if the pattern user must enter a value for the parameter. Mandatory parameters also display a field prompt to guide the pattern user.	
Field promot: Enter your parameter value	
Help Text (HTML) Enter any HTML or text that you want to display as help text for this parameter. Preview parameter help Do not include any <html> or <head> tags because the text is inserted into a parameter HTML file.</head></html>	
Describe the parameter here Describe the parameter here	

10. Make similar changes to the other parser pattern parameter IDs. Set them to MessageModel, Message and PhysicalFormat.

At this point, the pattern configuration should look like this. Note that the Parameter IDs for each of the properties is shown in brackets, following the name of the property.



11. Now we are going to create expressions which will dynamically enable or disable parser properties, based on the selected Message Domain.

Select the Message Domain property, and click Edit.

Select the Editor tab. You will see that the "Parameter editor" has automatically been set to Drop Down Selection, and the default values are restricted to MRM and XMLNSC.

Set the default value to MRM.

Click OK.

🖹 Edit Parameter: Message domain		X
Configure pattern parameter		
Configure the pattern parameter and how it	: is displayed to pattern users.	
Basic Edico' Transform Enable		
Parameter Editor		
Parameter editor:	Drop Down Selection Configure Editor	
Type selection:	Message domain Enumerated Types Tables	
Default vaue:	MRM : For binary or text messages that are modeled in a message set	
	XMLNSC : For XNL messages (namespace aware, validation, low memory use)	
Dependencies	MRM : For binary or text messages that are modeled in a message set	
Pattern parameters can depend on one or the message types that are available in the	r more parameters. For example, a message type parameter depends on a message set parameter because it displays re selected message set.	
Dependencies are configured automatical	ly by the Pattern Authoring editor.	
To ensure dependencies are configured o	orrectly, the dependent parameter (for example, the message type parameter) must be in the same group and be	
arter the parameter on which it depends.		
This parameter depends on the following	parameters:	

12. If the pattern user selects XMLNSC, then you will want to disable the option of selecting the MRM parser properties (message model, message and physical format). We will do this by constructing an XPath expression for the other parser parameters.

Double-click the Message Model property, to edit it..



13. Select the Enable tab.

Using the Expression field, we will construct an XPath expression. This will enable or disable the Message Set property, based on the value of the Message Domain parameter entered by the pattern user.

If the expression evaluates to "true", then the field will be displayed to the pattern user. If it evaluates to false, it will not be displayed.

Edit Parameter: Message model				2
Configure pattern parameter				
Configure the pattern parameter and how it is displayed to	pattern users.			
a late la c Fashial				
Basic Editor Transform Enable Configure an XPath expression that controls whe If the expression evaluates to true, the parameter Disabled parameters do not change target proper	n this parameter is enabled in the Patt rr is enabled, otherwise it is disabled, ties when a pattern instance is gener/	ern Instance editor. ated.		
Functions		Operators		
B → t Boolean B → t Number B → t String Function name:	रु Use	-+ 		S Use
Groups and Parameters	Parameter ID	Tost Value		
Coups and Parameters Coups and Par	Parameter 10	Tiest value	5	
🗎 Message domain	MessageDomain	MRM		
 ≧ Message model ≌ Message ≅ Physical format 	MessageModel Message PhysicalFormat			
Test value:	👱 Set	Parameter ID:	MessageModel	Use
Expression Evaluation				
Expression:				Evaluate
Kesuit:				
				OK Cancel

14. Double-click the Message Domain field. This will populate the Expression field with

pp:getValue('MessageDomain')

Complete the expression by typing the value to compare the expression with, as follows:

pp:getValue('MessageDomain') = 'MRM'

This will Enable or Disable the Message Set property, based on the value of the Message Domain property.

dit Parameter: Message model				
nfiqure pattern parameter				
opfigure the pattern parameter and how it is displayed (to nattern users			
oningare the pattern parameter and now it is asplayed.	o pacernasers.			
sic Editor Transform Enable				
Configure an XPath expression that controls we If the expression evaluates to true, the parameters	en this parameter is enabled in the Pa eter is enabled, otherwise it is disabled	ttern Instance editor. I.		
 Disabled parameters do not change target prop 	erties when a pattern instance is gene	erated.		
unctions		Operators		
T				
E → ta boolan		-		
🗄 📲 Pattern		*		
🗄 📲 String		div		_
Function comes	R_ 1/2	Consultant		
		operator.	1	0.000
Pattern Parameters				
Groups and Parameters	Parameter ID	Test Value		1
Application Queue Information				
🖃 🔃 MQ_Input Node Parser Properties				
🖹 Message domain	MessageDomain	MRM		
🖹 Message model	MessageModel			
🖹 Message	Message			
🖹 Physical format	PhysicalFormat			
Testudios		Deven also TD	Manager	
Test value:	<u>~ Sec</u>	Parameter ID:	MessageDomain	
Expression Evaluation				
Expression	us/MassageDensis") 'MDM'			🖿 Evaluate
pp:getvai	ac(messageDumain) = mkm			Produce
Result:				
1				
				OK Cance

15. Check the XPath expression by using the Evaluate button. Here is the expected output.

Use the "Test value" field and the Set button to check correct operation of the expression.

Groups and Parameters	Parameter ID	Test Valu	e	
표 💽 Application Queue Information				
🖃 🐻 MQ_Input Node Parser Proper	ties			
🚔 Message domain	MessageDomain	MRM		
🗎 Message model	MessageModel			
🖹 Message	Message			
管 Physical format	PhysicalFormat			
est value: MRM	😤 Set	Parameter ID:	MessageDomain	😺 Use
pression Evaluation				
pression Evaluation	pp:getValue('MessageDomain') = 'MRM'			► Evaluate
pression Evaluation (pression: asult:	pp:getValue(MessageDomain') = 'MRM' Enabled (true)			Evaluate
pression Evaluation pression: esult:	pp:getValue('MessageDomain') = 'MRM' Enabled (true)			Evaluate

16. Check to see what happens if the domain is XMLNSC. In the "Test value" field, enter XMLNSC and click Set.

Click Evaluate. The expected output this, where the result is "Disabled(false)". In this case, the message set field would not be shown.

Groups and Parameters	Parameter ID	Test Value			
🗄 🔚 Application Queue Information					
🖃 🔚 MQ_Input Node Parser Propertie	s				
🚔 Message domain	MessageDomain	XMLNSC			
🖹 Message model	MessageModel				
🚔 Message	Message				
管 Physical format	PhysicalFormat				
extransion Evaluation	<u> </u>	Parameter ID:	MessageDomain		se
pression Evaluation					
(pression:	pp:getValue('MessageDomain') = 'MRM'			📄 Evaluate	
esult:	Disabled (false)				

Before we leave this parameter, copy the value in the Expression field to the clipboard (Ctrl-C); we will use it for the remaining parser properties.

Click OK to complete the Message Model properties.

17. Now repeat the same configuration for the Message and Physical Format properties (steps 11 to 16).

Highlight each property in turn, click Edit, select the Enable tab, and paste the contents of the clipboard into the Expression field.

Click OK.

Save the updated pattern.

18. You're done with pattern configuration. Now you need to test the updated pattern.

On the "Create Pattern" tab, click Create Pattern Plug-ins.

When this is complete, click Test Pattern.

📧 Transform.msgflow 🛛 🔡 I	MyPattern.pattern 🛛
进 Create Pattern	
Test your pattern by configurin	ng your pattern plug-in information, click "Test Pattern".
A The plug-ins in your works	space may be out of date and need to be re-created.
🛃 Create Pattern Pl	ug-ins 🚺 Test Pattern
Plug-in Information	
🛃 Configure the unique ide	entifier for your pattern plug-in.
Pattern name:	MyPattern
Plug-in ID:	com.betaworks.ibm.domain.MyPattern
Version:	1.0.0.0
Provider:	IBM BetaWorks
Description:	Plug-in created by BetaWorks Early Programs
Translation Options	
If you enable this option, the If you are creating a single la	Pattern Authoring editor creates two additional NLS plug-ins. These anguage pattern, do not select this check box.
Create translation plug-in	is (*.nl1 and *.doc.nl1)
Source Files Pattern Configuration	on Categories Create Pattern

19. Accept the location of the second workspace.

20. If not automatically selected, click on Patterns Explorer, and click MyPattern.



21. Click "Create New Instance", provide a name, and click OK.

Wew Pattern Instance	<u>_ 🗆 ×</u>
Create New Instance The instance name uniquely identifies the pattern instance project that is created when the pattern instance is saved or generated. The name ideally reflects the business and integration functions that the pattern supports.	
Pattern instance name: ParserDemo	
ОК	Cancel

22. Expand the MQ Input Node Parser Properties group.

The default value for Message Domain is MRM. In this case, the Message model, message and physical format parameters must be provided.

For this pattern to be generated, an appropriate message set must be present in the developers workspace, so this is the limit of the part of the task.

📲 Pattern Specification 🛛 🔒	B ParserDemo - Pattern Configuration 🕱			
🐁 Configure Patte	rn Parameters			
Provide values for pattern para	meters. Click the "Generate" button or click <u>here</u> to generate a pattern instance.			
O The pattern parameter "Mes	sage model" is mandatory but a value is not set.			
Pattern Parameters		Ŧ		ž
Application Queue Info	ormation		V	
▼ MQ_Input Node Parser	Properties		0	3
Pattern Parameters				
Message domain *	MRM : For binary or text messages that are modeled in a message set		•	
Message model *	<enter parameter="" value="" your=""></enter>		•	
Message *	<enter parameter="" value="" your=""></enter>		•	
Physical format *	<enter parameter="" value="" your=""></enter>		•	
L				
Generate				
Specification Configuration				

23. On the other hand, if the Message Domain is set to 'XMLNSC', then the remaining parser properties are greyed out, and will not be specified.

In this case, you can proceed to generate a new pattern instance. Use the same approach as in the earlier labs, and observe the resulting MQ Input node properties in the generated message flow.

📲 Pattern Specification 🛛 🔒	*ParserDemo - Pattern Configuration 🕱		
🌯 Configure Patter	n Parameters		
Provide values for pattern parar	neters. Click the "Generate" button or click <u>here</u> to generate a pattern instance.		
i Pattern parameters are read	y. Click the "Generate" button to generate a pattern instance.		
Pattern Parameters		(= 🗳
Application Queue Info	rmation		
▼ MQ_Input Node Parser	Properties		
Pattern Parameters			
Message domain *	XMLNSC : For XML messages (namespace aware, validation, low memory use)		•
Message model *	<enter parameter="" value="" your=""></enter>		V
Message *	<enter parameter="" value="" your=""></enter>		$\overline{\mathbf{v}}$
Physical format *	<enter parameter="" value="" your=""></enter>		V
Generate			
Specification Configuration			

This concludes the Pattern Authoring Parser Properties lab.