WebSphere Message Broker Version 7.0.0.1

Pattern Authoring Lab 3

Setting parser properties with XPath expressions

September, 2010

Version 1.0

Hands-on lab built at product code level version 7.0.0.1

1. Lab Objectives

In this lab, you will see how to configure the parser properties on an MQInput node. Configuring the message domain, set, type and 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 set, format and type 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 Message Broker 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 Broker 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.pat	ttern	- E
👌 🔅 Palette			_
Selection]		
Connection			
Note			
🕞 Favorites			
R WebSphere MQ	88		
Gins JMS			
	MQ In		
🧟 Web Services			
🔃 SCA			
🖓 WebSphere Adapters 📃	J		
🕞 Routing			
😕 Transformation 🛛 🗠			
Co Database			
😽 File 🗸	1	Log queue ouchor	
Graph User Defined Propertie	es		
🔲 Properties 🛛 🖹 Pro	blems 🔲 🗖 Deploym	nent Log) 📮 Console) 🗸 🗸	- E
🛃 MQ Input Node Pro	perties - MQ Inj	put	
		•	
Description Basic	Message domain	<select a="" blob="" default="" domain="" in="" is="" message.="" no="" present="" the="" to="" use="" when=""></select>	•
Input Message Parsing	-	Consection domain to use which no domain is present in the message, behavior is buody	
Parser Options	Message set		•
Advanced	Message type		•
Validation	Message format		-
Security			
Instances			

This shows the message parsing properties of the MQ Input node. The default message domain is XMLNSC. 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.

🕫 Transform.msgflow 🔀 🔡 MyPattern.p	pattern		
👌 😳 Palette			
Selection			
🖉 Note			
🙀 Favorites			
🖳 WebSphere MQ		_	
Ста змя	Ĵġ───►¤́¢≱───►¤́₫		
С НТТР МО	Input) Transform MQ O	utput	
强 Web Services	Select Target Properties 📀 🕱		
🔍 SCA			
🖓 WebSphere Adapters	E Advanced		
GRouting			
🔁 Transformation 🗠	Queue name		
	Minder Message Parsing	ab	
CoDatabase	Message format		
🕞 File 🚽	Message set Log qu	ieue output	
Graph User Defined Properties			
🔲 Properties 🔀 [Problems] 🌐 Deployment Log 📃 Console			
😰 MQ Input Node Properties - MQ Input			

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).

Transform.msgflow	
😂 Source Files	
Select the source files to include in your pattern. You can also view the target properties a To select target properties, select your source file and click "Select Target Properties", or or Then right-click a node or the canvas and select "Patterns > Select Target Properties".	
Select Source Files	Target Properties
type filter text	type filter text
Image: Second secon	Transform.mqsi.Transform MQ_Input Transform.mqsi.Transform.MQ_Input.messageDomainProperty Transform.mqsi.Transform.MQ_Input.messageSetProperty Transform.mqsi.Transform.MQ_Input.messageSetProperty Transform.mqsi.Transform.MQ_Input.messageTypeProperty Transform.mqsi.Transform.MQ_Input.queueName
Source Files Pattern Configuration Categories Create Pattern	

4. Switch to the Pattern Configuration tab.

> 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
- 2. Message Set
- Message Format
 Message Type

ansform.msgflow 🔐 *MyPattern.pattern 🖄 Pattern Configuration figure your groups and pattern parameters and associate the pattern parameters with their target properties. s tree shows the targets that are set by each pattern parameter. u can also configure Java and PHP code which is invoked when a pattern instance is created.	
bupps and Parameters Image: Mo_Input Image: Mo_Input Image: Sets property: Transform.mgsi.Transform.MQ_Input.queueName Image: Queue parefix (pp2) Image: Queue parefix (pp3) Image: Message domain (pp4) Image: Queue parefix (pp5) Image: Property: Transform.mgsi.Transform.MQ_Input.messageFormatProperty Image: Property: Transform.mgsi.Transform.MQ_Input.messageTypeProperty Image: Property: Property: Tran	

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.

🧓 Edit Group: MQ_Inpul		×	
Configure group Configure the pattern p Enter a group display na	arameter group and how it is displayed to pattern users. me and a description.		
Display name:	MQ_Input Node Parser Properties		
Description:	Pattern Parameters		
Group Options			
🔽 Generate help docum	entation Select this option to create help the pattern.	information for this group in	
Display this group	Select this option to make this o the pattern user.	Select this option to make this group visible in the pattern to the pattern user.	
		OK Cancel	

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

# *MyPattern.pattern 🛛	
E Pattern Configuration	
Configure your groups and pattern parameters and associate the pattern parameters with their target properties. This tree shows the targets that are set by each pattern parameter. You can also configure Java and PHP code which is invoked when a pattern instance is created. Groups and Parameters Imperime Application Queue Information	Add Group
Euler enter (queueName)	Add Parameter
—	Edit
Image: Compute Node Parser Properties	Delete
	Enumerated Types
Sets property: Transform.mqsi.Transform.MQ_Input.messageSetProperty	8 8
Sets property: Transform.mqsi.Transform.MQ_Input.messageTypeProperty	
Java and PHP Code	
	Add
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).

Configure the enumerated types that are used by the pattern parame	ers.	
n enumerated type is a set of values that a pattern parameter accepts he display names are shown in a list for pattern users when they create he value is the content that is stored in the target property and can be		
Enumerated type: Message domain		
	Add Re	move Rename Duplicate
Display Name	Value	
MRM : For binary, text or XML messages (namespace aware, valida	MRM	Add
MLNSC : For XML messages (namespace aware, validation, low me	XMLNSC	
DataObject : For data from WebSphere Adapters, CORBA and Data (MLNS : For XML messages (namespace aware)	DataObject XMLNS	Remove
MSMap : For JMS MapMessage messages (XML)	JMSMap	
IMSStream : For JMS StreamMessage messages (XML)	JMSStream	
ISON : For JavaScript Object Notation messages	JSON	
MIME : For MIME wrapped data including multipart	MIME	
3LOB : For messages with an unspecified format	BLOB	
(ML : For XML messages (deprecated - use XMLNSC)	XML	
DOC : For SAP ALE IDocs from the WMQ Link for R/3 (deprecated	IDOC	
		Reset Values
ou cannot remove an enumerated type if it is being used by a paramete nange the type of a pattern parameter by clicking Edit in the Pattern Pa nis enumerated type is used by the following parameters:		

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 multiple lines from DataObject line to the bottom of the list (use upper-case and left-click, as in multiple selections in Windows Explorer), and click Remove.

Alternatively, just click Remove several times, and remove the unwanted lines one by one.

Click OK to return to the Pattern Configuration window.

Configure Enumerated Types		×	
Configure enumerated types Configure the enumerated types that are used by the pattern parameters.			
An enumerated type is a set of values that a pattern parameter accepts The display names are shown in a list for pattern users when they creat The value is the content that is stored in the target property and can be	e an instance of this pattern.		
Enumerated type: Message domain		•	
	Add Remove	Rename Duplicate	
Display Name MRM : For binary, text or XML messages (namespace aware, valida XMLNSC : For XML messages (namespace aware, validation, low me Dataobject : For data from WebSphere Adapters, CORBA and Databa XMLNS : For XML messages (namespace aware) JMSSMap : For JMS MapMessage messages (XML) JMSStream : For JMS StreamMessage messages (XML) JMSStream : For JMS StreamMessage messages (XML) JMSStream : For JMS StreamMessage messages (XML) JSON : For JavaScript Object Notation messages MIME : For MIME wrapped data including multipart BLOB : For messages (deprecated - use XMLNSC) IDOC : For SAP ALE IDocs from the WMQ Link for R/3 (deprecated	XMLNS JMSMap JMSStream JSON MIME BLOB XML	Add Remove	
You cannot remove an enumerated type if it is being used by a parameter or is defined by a target property. Change the type of a pattern parameter by clicking Edit in the Pattern Parameters tab.			
This enumerated type is used by the following parameters: □ □ □ MQ_Input □ □ □ Message domain □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □			

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.

Edit Parameter: Message domain	×			
onfigure the pattern parameter				
Configure the pattern parameter and how it is displayed to pattern users.				
Basic Editor Transform Enable				
Display name: Message domain Parameter ID: MessageDomain				
Parameter Options				
E 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.				
Configure during deployment Select this option if the parameter maps to a target property and you want the pattern user to override it in the BAR file.				
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 prompt: 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>				
Cep>Describe the parameter here Describe the parameter here				

10. Make similar changes to the other parser pattern parameter IDs. Set them to MessageSet, MessageFormat and MessageType.

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.

🔢 Transform.msgflow 🔡 *MyPattern.pattern 🛛				
© Pattern Configuration				
Configure your groups and pattern parameters and associate the pattern parameters with their target properties.				
This tree shows the targets that are set by each pattern parameter.				
You can also configure Java and PHP code which is invoked when a pattern instance is created.				
Groups and Parameters				
Groups and Parameters				
🗆 💽 MQ_Input	Add Group			
Queue name (pp1)				
└────────────────────────────────────	Add Parameter			
	Edit			
E Quode santx (pp) ⊡ E Message domain (MessageDomain)	Latin			
Sets property: Transform.mqsi.Transform.MQ_Input.messageDomainProperty	Delete			
⊟ 😫 Message set (MessageSet)				
└── Sets property: Transform.mqsi.Transform.MQ_Input.messageSetProperty	Enumerated Types			
	2 3			
En Ses property, mansforming, mansforming_input inessager of mathroperty				
Sets property: Transform.mqsi.Transform.MQ_Input.messageTypeProperty				
Source Files Pattern Configuration Categories Create Pattern				

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		×
Configure the pattern paramete	r	
Configure the pattern parameter and ho	w it is displayed to pattern users.	
Basic Editor Transform Enable		
Parameter Editor		
Parameter editor:	Drop Down Selection	
Enumerated type:	Message domain Enumerated Types	
Default value:	MRM : For binary, text or XML messages (namespace aware, validation, low memory use)	
	MRM : For binary, text or XML messages (namespace aware, validation, low memory use) XMLNSC : For XML messages (namespace aware, validation, low memory use)	
Dependencies	XMENDE : For XML messages (namespace aware, validation, low memory use)	
Dependencies are configured automat To ensure dependencies are configure after the parameter on which it depen	d correctly, the dependent parameter (for example, the message type parameter) must be in the same group and be	
This parameter depends on the followi	ng parameters:	

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

Highlight the Message Set property, and click Edit.

🖽 Transform.msgflow 🔠 *MyPattern.pattern 🛛	
Pattern Configuration MyPattern/MyPattern.pattern	
Configure your groups and pattern parameters and associate the pattern parameters with their target properties. This tree shows the targets that are set by each pattern parameter. You can also configure Java and PHP code which is invoked when a pattern instance is created. Groups and Parameters	
Image: Mog_Input Image: Sets property: Transform.mqsi.Transform.MQ_Input.queueName Image: Queue suffix (pp2) Image: Queue suffix (pp3) Image: Property: Transform.mqsi.Transform.MQ_Input.messageDomainProperty Image: Property: Transform.mqsi.Transform.MQ_Input.messageSetProperty Image: Property: Transform.mqsi.Transform.MQ_Input.messageFormatProperty Image: Property: Transform.mqsi.Transform.MQ_Input.messageTypeProperty Image: Property: Transform.mqsi.Transform.MQ_Input.messageTypeProperty Image: Property: Transform.mqsi.Transform.MQ_Input.messageTypeProperty Image: Property: Transform.mqsi.Transform.MQ_Input.messageTypeProperty	Add Group Add Parameter Edit Delete Enumerated Types
Source Files Pattern Configuration Categories Create Pattern	

13. Select the Enable tab, and highlight the Message Set parameter.

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 set			>
Configure the pattern parameter			
Configure the pattern parameter and how it is displayed to patt	ern users.		
Basic Editor Transform Enable			
Configure an XPath expression that controls when this	parameter is enabled in the	Pattern Instance editor.	
If the expression evaluates to true, the parameter is a	enabled, otherwise it is disab	led.	
Functions		Operators	
		· · · +	
ter>te Number			
E Pattern		div	
⊞			
		···· !=	
Function name:	묷 Use	Operator:	To Use
, <u> </u>			
Pattern Parameters			
Groups and Parameters	Parameter ID	Test Value	
🖂 🗔 MQ_Input			
Queue name	pp1	queue	
 Queue prefix ↔ 	pp2	qp.	
 	pp3 MessageDomain	.qs MRM	
	MessageSet	МКМ	
🚔 Message format	MessageFormat		
Test value:	👱 Set 🛛 Pa	rameter ID: MessageSet	🐺 Use 🛛
Expression Evaluation			
Expression:			▶ <u>E</u> valuate

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.

Edit Parameter: Message set				×
Configure the pattern parameter				
Configure the pattern parameter and how it is displayed to pattern u	isers.			
and the second second second				
Basic Editor Transform Enable				1
Configure an XPath expression that controls when this para If the expression evaluates to true, the parameter is enable	ameter is enabled in the ed. otherwise it is disabl	Pattern Instance edito ed	pr.	
Functions		Operators		
T → → ta Boolean		· · · · · · · · · · · · · · · · · · ·		
terreta Boolean terreta Number		+		<u> </u>
E → → ta Number		*		
±		div		
				-
· · · · · · · · · · · · · · · · · · ·				
Function name: pp:getValue	Use	Operati	or:	रु Use
- Pattern Parameters				
			1	
Groups and Parameters	Parameter ID		Test Value	
□ E MQ_Input	pp1		CUOU0	
Queue prefix	pp1 pp2		queue qp.	
Queue suffix	pp3		,qs	
😫 Message domain	MessageDomain		MRM	
🖹 Message set	MessageSet			
	MessageFormat			-
	0			
Test value: MRM	😤 Set 🛛 Par	ameter ID: Me	essageDomain	Use
Expression Evaluation				
Expression: pp:getValue('MessageDomain') = 'MRM'				▶ <u>E</u> valuate

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 Value	<u> </u>
🖃 💽 MQ_Input			
🚔 Queue name	pp1	queue	
😫 Queue prefix	pp2	qp.	
🚔 Queue suffix	ррЗ	.qs	
🖹 Message domain	MessageDomain	MRM	
🖹 Message set	MessageSet		
😂 Message format	MessageFormat		•
est value: MRM	Set Paramete	er ID: MessageDomain	🐺 Use
xpression Evaluation			
xpression: pp:getValue('MessageDomain')	= 'MRM'		► 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.

	5	Parameter ID		Test Value	<u> </u>
🖂 💽 MQ_Input					
管 Queue nam	e	pp1		queue	
📔 Queue pref	ix	pp2		qp.	
管 Queue suff	ix	ррЗ		.qs	
🗎 Message de	omain	MessageDomain		XMLNSC	
管 Message se	et	MessageSet			
🚆 Message fo	rmat	MessageFormat			-
est value: XMLN xpression Evaluation	SC	Set	Parameter ID:	MessageDomain	Use
xpression:	pp:getValue('MessageDomain') = 'MR	M			Evaluate

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

Click OK to complete the Message Set properties.

17. Now repeat the same configuration for the Message Format and Message Type properties (steps 11 to 15).

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

Click OK.

18. You're done with pattern configuration. Now you need to rebuild the pattern plug-ins.

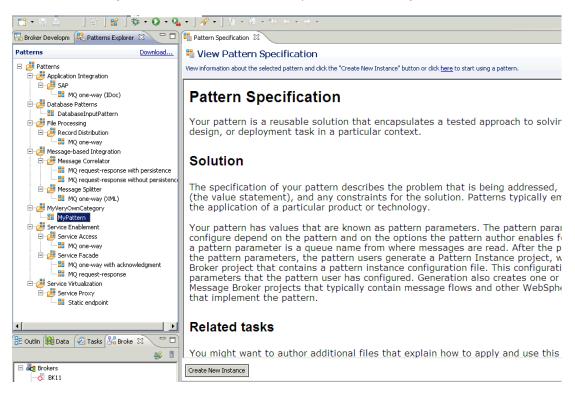
On the "Create Pattern" tab, click Create Pattern Plug-ins.

When this is complete, click Launch workbench.

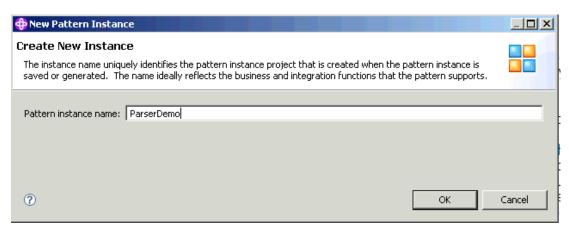
💷 Transform.msgflow	B MyPattern.pattern 🛛	
进 Create Pat	tern	
Test your pattern b	y configuring your pattern plug-in information, click "Create Pattern Plug-ins", and click "Launch Wor	kbench".
🔥 The plug-ins in	your workspace may be out of date and need to be re-created.	
Plug-in Informati	on	
🛃 Configure the	e unique identifier for your pattern plug-in.	🔗 Create Pattern Plug-ins
Pattern name:	MyPattern	🖶 Launch Workbench
Plug-in ID:	çom.your.company.domain.MyPattern	📲 Create Pattern Archive
Version:	1.0.0.0	
Provider:	Your Company Name	
Description:	Plug-in created by the Pattern Authoring editor	
Translation Optio	ns	
up so that you ca	option, the Pattern Authoring editor creates two additional NLS plug-ins. These plug-ins are set n drop in translated resources, such as Java property files. If you are creating a single language lect this check box.	
Create transla	tion plug-ins (*.nl1 and *.doc.nl1)	
Pattern Distribut	ion	
package your pat	reated and tested your pattern plug-ins (see the Plug-in ID above for the plug-in names), tern by clicking "File > Export > General > Archive File" to export these plug-ins.	
	io plug inc. the pattern user much extract the expected archive files into the default WebSphere	

19. Accept the location of the second workspace.

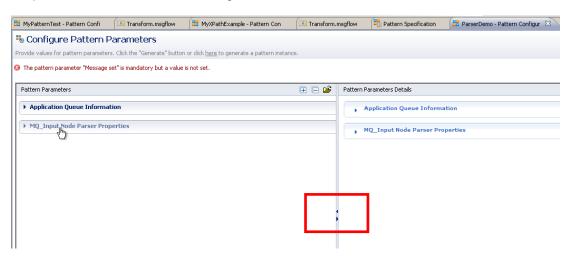
20. If not automatically selected, click on Patterns Explorer, and click MyPattern.



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



22. First, to see the full details, click on the small arrow on the centre pane divider to expand the left pane, or slide the divider to the right



Giving

🖥 MyPatternTest - Pattern Confi	📧 Transform.msgflow	🚦 MyXPathExample - Pattern Con	🕒 Transform.msgflow	Pattern Specification	🔡 ParserDemo - Pattern Configur 🛛 🕄
🗞 Configure Pattern P	arameters				
rovide values for pattern parameters	. Click the "Generate" buttor	n or click <u>here</u> to generate a pattern instar	ice.		
The pattern parameter "Message set in the pattern parameter set in the pattern pattern parameter set in the pattern patte	et" is mandatory but a value	is not set.			
Pattern Parameters				E 🖻 🖻	Pattern Parameters Details
Application Queue Informat	ion				, Application Queue
MQ_Input Node Parser Prop	erties			0	MQ_Input Node Parser
					5

23. Expand the MQ Input Node Parser Properties group.

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

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

🔡 MyPatternTest - Pattern Confi	Transform.msgflow	🔡 MyXPathExample - Pattern Con	💷 Transform.msgflow	Pattern Specification	
🖏 Configure Pattern Parameters					
Provide values for pattern parameter	s. Click the "Generate" button o	or click <u>here</u> to generate a pattern instance			
8 The pattern parameter "Message :	set" is mandatory but a value is	not set.			
Pattern Parameters				🕀 🖻 🎽	
Application Queue Information	tion				
▼ MQ_Input Node Parser Pro	perties			0	
Pattern Parameters					
Message domain *	MRM : For binary, text or XML	messages (namespace aware, validation, l	low memory use)		
Message set *	<enter parameter="" value="" your=""></enter>	>			
Message format *	<enter parameter="" value="" your=""></enter>	5		•	
Message type *	<enter parameter="" value="" your=""></enter>	>			

24. 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 Patte	ern Parameters	
Provide values for pattern para	ameters. Click the "Generate" button or click <u>here</u> to generate a pattern instance.	
i Pattern parameters are rea	dy. Click the "Generate" button to generate a pattern instance.	
Pattern Parameters		⊞ 🖻 🗳
▼ MQ_Input		
Pattern Parameters		
Queue name *	queue	
Queue prefix *	qp.	
Queue suffix *	.qs	
Message domain *	XMLNSC : For XML messages (namespace aware, validation, low memory use)	
Message set *	<enter parameter="" value="" your=""></enter>	V
Message format *	<enter parameter="" value="" your=""></enter>	
Message type *	<enter parameter="" value="" your=""></enter>	Y

This concludes the Pattern Authoring Parser Properties lab.