### WebSphere Message Broker Version 7.0.0.1

# Pattern Authoring Lab 2

## Extending Pattern Authoring with XPath

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 pattern parameters using the XPath expression tools in the pattern builder.

The MQ Input node in this lab has a 3-part queue name, comprising a queue prefix, the main part of the queue name (in the middle), and a queue suffix. Each of these three components will be defined as a pattern parameter. When a pattern instance is generated, the value of the pattern parameters will be concatenated using an XPath expression. The resulting string from the concatenation will be used to configure the queue name target property.

The lab goes on to make the prefix and suffix components "read only". This means that the pattern user need only be concerned with the specific application requirements for this message flow.

The starting point is the "Pattern Parameters" tab in the Pattern Authoring editor. This lab follows on from the first lab, Pattern Authoring Introduction, which created a pattern with a single parameter, "Queue Name".

#### 2. Extending the Pattern from the Basic Lab

1. Start this lab with the Pattern Authoring editor for the pattern "MyPattern.pattern", as used in the first 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 "Pattern Configuration" tab, then click the "Add Parameter" button.



2. The Add Parameter window will open.

· · · · · · · · · · · · · · · · · · ·	
igure the pattern parameter	
nfigure the pattern parameter and how i	t is displayed to pattern users.
Editor Transform Enable	
Varameter Display	Darameter ID: pn2
arameter 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.
<ul> <li>Configure during deployment</li> </ul>	Select this option if the parameter maps to a target property and you want the pattern user to override it in the BAR file.
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: Epter your para	meter value
lelp Text (HTML)	display as help text for this parameter. Preview parameter help
lelp Text (HTML) inter any HTML or text that you want to so not include any <html> or <head> ta arameter HTML file.</head></html>	display as help text for this parameter. Preview parameter help gs because the text is inserted into a
Find point [Find your para leip Text (HTML) inter any HTML or text that you want to o not include any <html> or <head> ta arameter HTML file.</head></html>	display as help text for this parameter. Preview parameter help gs because the text is inserted into a Describe the parameter here
Find your para relip Text (HTML) inter any HTML or text that you want to bo not include any <html> or <head> ta arameter HTML file.</head></html>	display as help text for this parameter. Igs because the text is inserted into a
Enter your para leip Text (HTML) inter any HTML or text that you want to Do not include any <html> or <head> ta arameter HTML file.</head></html>	display as help text for this parameter. gs because the text is inserted into a  Describe the parameter here

3. On the Basic tab:

Change the Display Name to "Queue prefix".

Change the Parameter ID field to "queuePrefix" (the default is something like "pp2").

Change the "Field prompt" to "Enter your queue prefix". This is the field that is highlighted in pink when the pattern user generates a new pattern instance. This helps the pattern user choose appropriate values for these properties.

Change the text description of the field by editing the HTML text window. The output version of the html is shown simultaneously in the pane on the right.

🚔 Add Parameter						×
Configure the patte	ern parameter					
Configure the pattern p	parameter and how it is dis	played to pattern users.				
Basic Editor Transf	iorm Enable					
- Parameter Display -						
Display name:	Queue prefix	Parameter	ID:	queuePrefix		
Parameter Options						
F Hide the paramet	ter	Select this option to hide the paramel when a pattern instance is created.	er and to use an	n XPath expression to set th	ne value of the param	eter
Configure during	deployment	Select this option if the parameter ma the BAR file.	ips to a target pr	roperty and you want the p	attern user to overrio	de it in
Mandatory paran	neter	Select this option if the pattern user i a field proop to guide the pattern us	nust enter a valu ;er.	ue for the parameter. Mand	latory parameters als	o display
Field prompt:	Enter your queue pre	Fix				
- Help Text (HTML) -						
Enter any HTML or te Do not include any <	ext that you want to displa (html> or <head> tags beg</head>	y as help text for this parameter. ause the text is inserted into a	Preview parame	eter help		
parameter HTML file.						
The first part of the second secon	of the queue name					<u> </u>
			The first pa	art of the queue name		
		_				
		<u></u>				<b>_</b>
						Cancel
						Cancer

4. On the Editor tab:

The Editor tab is where you configure the parameter editor for this pattern parameter. The parameter editor is displayed in the Pattern Instance editor. The default for a pattern parameter is a standard text entry editor.

Set the default value for the Queue Prefix to "qp." (do not include the quotation marks).

figure the pattern parameter and how it is displayed to pattern users.         c       Editor       Transform       Enable         arameter Editor	figure the pattern parameter and how it is displayed to pattern users.     c   E   c   E   c   E   c   E   c   C   E   c   C   E   c   C   E   C   C   E   C   C   E   C   C   E   C   C   E   C   C   C   E   C   <	figure the pattern parameter and how it is displayed to pattern users.         c       Editor       Transform       Enable         tarameter Editor       tarameter Editor         tarameter editor:       Text Editor         tarameter ditor:       Text Editor         tarameter dupe:       Image: I	figure the pattern parameter and how it is displayed to pattern users.         c       Editor         Transform       Enable         arameter Editor	igure the Pattern Par	ameter
Editor Transform Enable arameter Editor Text Editor Text Editor Enumerated type: Enumerated type: Enumerated type: Improvement of the parameter set of the parameter set of the parameters are configured automatically by the Pattern Authoring editor.	Editor Transform Enable arameter Editor Text Editor Text Editor Enumerated type: Enumerated Types efault value: qp. Enumerated Types arameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. eependencies are configured automatically by the Pattern Authoring editor. O ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same roup and be after the parameter on which it depends.	Editor Transform Enable arameter Editor Text Editor Text Editor Text Editor Inumerated type: Text Editor Inumerated type: Intervent In	Editor Transform Enable arameter Editor arameter Editor Text Editor Text Editor Enumerated type: Enumerated type: Enumerated type: Improvement of the parameter of the parameter of the parameters of the parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. Example, a message type parameter depends on a message set parameter of the parameter of the parameter on which it dependent parameter (for example, the message type parameter) must be in the same roup and be after the parameter on which it depends.	figure the pattern paramete	r and how it is displayed to pattern users.
arameter Editor arameter editor: Text Editor numerated type: Pefault value: Pefault value: Pependencies Pependencies Pependencies Pependencies Pependencies are configured automatically by the Pattern Authoring editor. Pependencies are configured automatically by the Pattern Authoring editor. Pependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam Pependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam Pependencies are configured correctly. Pependencies are config	arameter Editor arameter editor: Text Editor numerated type: Enumerated type: efault value:  pendencies attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. ependencies are configured automatically by the Pattern Authoring editor. o ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam roup and be after the parameter on which it depends. his parameter depends on the following parameters:	arameter Editor arameter editor: Text Editor numerated type: efault value:	arameter Editor arameter editor: Text Editor numerated type: Enumerated type: Enumerated Types efault value: gp.  ependencies attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set.  rependencies are configured automatically by the Pattern Authoring editor. o ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same roup and be after the parameter on which it depends.  his parameter depends on the following parameters:	Editor Transform En	able
arameter editor:       Text Editor         inumerated type:       Image: Enumerated Type:         befault value:       Image: Imag	arameter editor:       Text Editor         inumerated type:       Image: Enumerated Types         befault value:       Image: Imag	arameter editor: Text Editor Inumerated type: Inumerated type: Inumerated type: Inumerated type: Inumerated type: Inumerated types parameters (Inumerated Types) International (International (Internatio	arameter editor:       Text Editor         inumerated type:       Image: Enumerated Types         befault value:       Image: Imag	arameter Editor	
Enumerated type: Enumerated Type: Default value: qp. Dependencies Pattern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter because it displays the message types that are available in the selected message set. Dependencies are configured automatically by the Pattern Authoring editor. To ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam group and be after the parameter on which it depends.	Enumerated type: Enumerated Types Default value: qp. Default value: qp. Dependencies Pattern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter because it displays the message types that are available in the selected message set. Dependencies are configured automatically by the Pattern Authoring editor. To ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam group and be after the parameter on which it depends. This parameter depends on the following parameters:	Enumerated type: Inumerated Types Default value:	Enumerated type: Enumerated Types Default value:	Parameter editor:	Text Editor
Default value:	Default value:	Default value:	Default value:	Enumerated type:	Enumerated Types
Dependencies <sup>3</sup> attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter because it displays the message types that are available in the selected message set. Dependencies are configured automatically by the Pattern Authoring editor. Fo ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam proup and be after the parameter on which it depends.	Dependencies Pattern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter because it displays the message types that are available in the selected message set. Dependencies are configured automatically by the Pattern Authoring editor. Fo ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same proup and be after the parameter on which it depends. This parameter depends on the following parameters:	Dependencies Pattern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter because it displays the message types that are available in the selected message set. Dependencies are configured automatically by the Pattern Authoring editor. Fo ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same group and be after the parameter on which it depends. This parameter depends on the following parameters:	Dependencies Pattern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter because it displays the message types that are available in the selected message set. Dependencies are configured automatically by the Pattern Authoring editor. Fo ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same group and be after the parameter on which it depends. This parameter depends on the following parameters:	Default value:	 др.
ependencies attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. ependencies are configured automatically by the Pattern Authoring editor. o ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam 'oup and be after the parameter on which it depends.	ependencies attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. ependencies are configured automatically by the Pattern Authoring editor. o ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam roup and be after the parameter on which it depends. his parameter depends on the following parameters:	ependencies attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. ependencies are configured automatically by the Pattern Authoring editor. o ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same roup and be after the parameter on which it depends. his parameter depends on the following parameters:	ependencies attern parameters can depend on one or more parameters. For example, a message type parameter depends on a message set parameter ecause it displays the message types that are available in the selected message set. ependencies are configured automatically by the Pattern Authoring editor. o ensure dependencies are configured correctly, the dependent parameter (for example, the message type parameter) must be in the same roup and be after the parameter on which it depends. his parameter depends on the following parameters:		
his narameter depends on the following narameters:				ependencies are configured	automatically by the Pattern Authoring editor.
				Fo ensure dependencies are o group and be after the param	e following parameters:
				To ensure dependencies are o group and be after the param This parameter depends on th	configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam eter on which it depends.
				To ensure dependencies are o group and be after the param This parameter depends on th	configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam eter on which it depends.
				To ensure dependencies are o group and be after the param This parameter depends on th	configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam eter on which it depends.
				To ensure dependencies are o group and be after the param This parameter depends on th	configured correctly, the dependent parameter (for example, the message type parameter) must be in the sam eter on which it depends. e following parameters:

- 5. Perform a similar set of actions for Queue Suffix:
  - 1) Click "Add Parameter"
  - 2) On the Basic tab:
    - Set Display Name = "Queue suffix".
    - Set Parameter ID = "queueSuffix".
    - Set Field Prompt = "Enter your queue suffix".
    - Set Help text = "The last part of the queue name".
  - 3) On the Editor tab: Set Default value = ".qs".

Click OK.

6. Finally, let's go back and set a sensible value for the Parameter ID of the queue name, which was left at the default value (pp1).

Highlight the Queue name parameter, and click Edit (or double-click the parameter)

🔡 *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	
□       ■ Application Queue Information         □       ■ Queue name (pp1)         □       ■ Sets property: Transform.mqsi.Transform.MQ_Input.queueName         □       ■ Queue prefix (queuePrefix)         □       ■ Queue suffix (queueSuffix)	Add Group Add Parameter Edit Delete Enumerated Types S

Change the Parameter ID to queueName, and click OK.

🖹 Edit Parameter: Queue name				×		
Configure the pattern param	neter					
Configure the pattern parameter an	d how it is displayed to pattern u	sers.				
Basic Editor Transform Enable	]					
Parameter Display						
Display name: Queue	name	Parameter ID:	queueName			
- Demonstern Orekiene						
Parameter Options						
Hide the parameter	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	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.					
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 yo	our queue name					
Help Text (HTML) Enter any HTML or text that you Do not include any <html> or <he parameter HTML file.</he </html>	want to display as help text for th aad> tags because the text is ins	is parameter. Preview param erted into a	eter help			
Enter the main part of the q	ueue name	Enter the r	main part of the queue name	<u> </u>		

- 7. You have now defined three pattern parameters in the group Application Queue Information. The parameters are:
  - Queue name
  - Queue prefix
  - Queue suffix

You have also provided meaningful names for the Parameter IDs.

You will now configure an XPath expression which will concatenate these parameters.

Highlight the "Queue name" parameter and click the Edit button.

🗄 *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	
□       Application Queue Information         □       □         □       □         Sets property: Transform.mqsi.Transform.MQ_Input.queueName         □       □         □       □         Queue prefix (queuePrefix)         □       □ </th <th>Add Group Add Torametor Edit Delete Enumerated Types R</th>	Add Group Add Torametor Edit Delete Enumerated Types R

8. As before, this will open the "Configure Pattern Parameter" window.

ic Editor Transform Enable		
Parameter Display Display name:	Parameter ID: queueName	
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.	
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.	h
Mandatory parameter	Select this option if the pattern user must enter a value for the parameter. Mandatory parameters also disp a field prompt to quide the pattern user.	laγ
Line brouber   cricer your d	sue name	
Help Text (HTML) Enter any HTML or text that you want Do not include any <html> or <head> parameter HTML file.</head></html>	ue name o display as help text for this parameter. Preview parameter help ags because the text is inserted into a	
Help Text (HTML) Enter any HTML or text that you want Do not include any <html> or <head> parameter HTML file.</head></html>	aue name o display as help text for this parameter. ags because the text is inserted into a hame Enter the main part of the queue name	

9. On the Editor tab, specify the default value for the Queue Name as "queue".

🚔 Edit Parameter: Queue name		x			
Configure the Pattern Parameter					
Configure the pattern parameter and how	t is displayed to pattern users.				
Basic Editor Transform Enable					
Parameter Editor					
Parameter editor:	Text Editor				
Enumerated type:	Enumerated type:  Enumerated Types				
Default value:	Default value: queue				
Pattern parameters can depend on one because it displays the message types the	r more parameters. For example, a message type parameter depends on a message set parameter lat are available in the selected message set.				
Dependencies are configured automatica	lly by the Pattern Authoring editor.				
To ensure dependencies are configured group and be after the parameter on wh	correctly, the dependent parameter (for example, the message type parameter) must be in the same ich it depends.				

10. Switch to the Transform tab.

The Transform tab allows you to configure an XPath expression which will transform the value of the pattern parameter. The expression is evaluated when the pattern instance is generated. The result of the evaluation updates the value of the pattern parameter.

In this exercise, you will configure an expression which concatenates the three pattern parameters into one string.

Expand the String functions, and double-click the "concat" function. This will populate the Expression field near the bottom of the window.

Note that since the expression is incomplete, you will see an error message at the top of the window.

dit Parameter: Queue name		
nfigure the pattern parameter		
An XPath expression for this parameter is inv	zalid	
asic   Editor Transform   Enable		
Configure an XPath expression to transform the The XPath expression can get the value of other	value of this parameter. The XPath expression is e parameters by using the getValue() XPath function	waluated when a pattern instance is created. n.
···· ··· ··· ··· ··· ··· ··· ··· ··· ·		
Functions	Oper	rators
⊟⊅ta String		+
tring		*
Concat		- div
		···=
substring-after		! <b>=</b>
substring actor		- <
Function name: concat	😽 Use	Operator: रु Use
Pattern Parameters		
Groups and Parameters	Parameter ID	Test Value
🖃 🔃 Application Queue Information		
🚔 Queue name	queueName	
Queue prefix	queuePrefix	qp.
👻 Queue suffix	queueSuffix	.qs
I		
Test value:	Z pet Paramete	the oregoing the o
Expression Evaluation		
Expression: concat()		🕨 Evaluate
Result:		
Result:		
Result:		

11. Double-click the "Queue prefix" parameter. This will add the parameter to the generated expression. The parameter is identified by the Parameter ID, queuePrefix, that you defined earlier.

Type a comma "," after the queue prefix name in the expression field.

The XPath expression is still invalid at this time, so you still see the message at the top of the window.

nfigure the pattern parameter		
Ap YBath expression for this parameter is inval	i-1	
asic Editor Transform Enable		
Configure an YBath expression to transform the up	alue of this parameter. The YBath expression is	evaluated when a nattern instance is created
The XPath expression can get the value of other p	arameters by using the getValue() XPath function	on.
Functions	Op	erators
Ė		r-+
		-
starts-with		
		!=
Substring artor		-<
Function name: concat	Use	Operator: 🐺 Use
Pattern Parameters		
Groups and Parameters	Parameter ID	Test Value
Groups and Parameters   Groups and Parameters	Parameter ID	Test Value
Groups and Parameters           Groups and Parameters           Image: Application Queue Information           Queue name	Parameter ID queueName	Test Value
Groups and Parameters □  □ Application Queue Information	Parameter ID queueName queuePrefix	Test Value qp.
Groups and Parameters □ I Application Queue Information	Parameter ID queueName queuePrefix queueSuffix	Test Value qp. .qs
Groups and Parameters □ I Application Queue Information	Parameter ID queueName queuePrefix queueSuffix	Test Value qp. .qs
Groups and Parameters □  □  □  □  □  □  □  □  □  □  □  □  □	Parameter ID queueName queuePrefix queueSuffix	Test Value       qp.       .qs
Groups and Parameters □  □ Application Queue Information Cueue name Cueue prefix Cueue prefix Cueue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value       qp.       .qs
Groups and Parameters  Groups and Parameters  Groups and Parameters  Queue Information  Queue name  Queue prefix  Queue prefix  Queue suffix  Test value:	Parameter ID queueName queuePrefix queueSuffix	Test Value       qp.       .qs
Groups and Parameters         Image: Construction Queue Information         Image: Queue name         Image: Queue prefix         Image: Queue suffix         Image: Queue suffix         Image: Test value:         Image: Queue qp.	Parameter ID queueName queuePrefix queueSuffix	Test Value       qp.       .qs       ter ID:     queuePrefix
Groups and Parameters         Groups and Parameters         Queue name         Queue prefix         Queue suffix         Queue suffix         Test value:         qp.         Expression Evaluation	Parameter ID queueName queuePrefix queueSuffix	Test Value       qp.       .qs       ter ID:     queuePrefix
Groups and Parameters Queue Information Groups and Parameters Queue prefix Groups and Parameters Queue prefix Groups and Parameters Queue suffix Test value:  Test v	Parameter ID queueName queuePrefix queueSuffix Set Paramet	Test Value       qp.       .qs
Groups and Parameters         Groups and Parameters         Queue name         Queue name         Queue prefix         Queue suffix         Test value:         qp.         Expression Evaluation         Expression:         concat(pp:getValue('q	Parameter ID queueName queuePrefix queueSuffix ueuePrefix'),∳	Test Value       qp.       .qs       ter ID:     queuePrefix       Use
Groups and Parameters	Parameter ID queueName queuePrefix queueSuffix Set Paramet ueuePrefix'),)	Test Value       qp.       .qs       ter ID:     queuePrefix       Use
Groups and Parameters         Groups and Parameters         Queue name         Queue name         Queue prefix         Queue suffix         Test value:         qp.         Expression Evaluation         Expression:         concat(pp:getValue('q         Result:	Parameter ID queueName queuePrefix queueSuffix Set Paramet ueuePrefix'),	Test Value       qp.       .qs       ter ID:     queuePrefix       Use
Groups and Parameters Queue Information Groups and Parameters Queue prefix Groups and Parameters Groups and Paramet	Parameter ID queueName queuePrefix queueSuffix Set Paramet ueuePrefix'),	qp. .qs :er ID:
Groups and Parameters         Image: Construction of the second	Parameter ID queueName queuePrefix queueSuffix Paramet ueuePrefix'),)	Test Value       qp.       .qs       ter ID:       queuePrefix       Image: Concel       OK

12. Double-click the "Queue name" parameter. This will concatenate the parameter to the XPath expression.

Type a comma "," as before.

🚔 Edit Parameter: Queue name			×
Configure the pattern parameter			
Configure the pattern parameter and how it is displayed to pa	ttern users.		
Basic Editor Transform Enable			(
Configure an XPath expression to transform the value of this	parameter. The XPath expressi	on is evaluated when a pattern instand	e is created.
The XPath expression can get the value of other parameters t	by using the getValue() XPath f	unction.	
Functions		Operators	
⊡+>ta String	<u> </u>	+	<u> </u>
Ca sering		*	
😌 starts-with		div	
	-	-<	
E serbine en ser	R 11m	Commission D	
Function name:   concat		Operator:	-th Ope
Pattern Parameters			
Crows and Davamakars	Davamatav TD	Test Value	
Application Queue Information	Parallieter 1D		
Queue name	queueName		
Queue prefix	queuePrefix	qp.	
	queueSuffix	.qs	
Test value:	Set Par	ameter ID:   queueSuffix	
Expression Evaluation			
	)D		
Expression:   concat(pp:getvalue( queuePrenx	),pp:getvalue( queuelvame ))		
Result:			
			OK Cancel

13. Double-click the "Queue suffix" parameter to complete the XPath expression.

Click the "Evaluate" button to make sure that the XPath expression logic is correct. The Evaluate function uses the default values for each pattern parameter.

<mark>÷</mark> Edit Parameter: Queue name			×
Configure the pattern parameter			
Configure the pattern parameter and how it is displayed to pattern	users.		
Basic   Editor Transform   Enable			1
Configure an XPath expression to transform the value of this param	eter. The XPath expression	n is evaluated when a pattern instance i	s created.
The xPath expression can get the value of other parameters by usi	ng the getvalue() xPath fi	Inction.	
Functions		Operators	
E → ting	<b>_</b>	+	<u> </u>
concat		*	
starts-with		div	
substring ator	•	<	<b>_</b>
Eunction name: concat	Use	Operator:	T Use
Pattern Parameters			
Groups and Parameters	Parameter ID	Test Value	
Application Queue Information			
	queueName queuePrefix	90	
	queueSuffix	,qs	
Test value:	😟 Set 🔰 Para	ameter ID: aueueSuffix	Use
		,	
Expression Evaluation			
Expression: concat(pp:getValue('queuePrefix'),pp:g	getValue('queueName'),pp	:getValue('queueSuffix'))	🕞 Evaluate
Becult			
Nosuki			
			OK Cancel

14. The test value can also be changed by clicking on the Set button, which highlights the required parameter.

Highlight the "Queue prefix" parameter, and set the Test value field to "QP.PRODUCTION.". Click Set.

Click Evaluate, and see the generate queue name in the Result field.

$\frac{2}{2}$ Edit Parameter: Queue name			X
Configure the pattern parameter			
Configure the pattern parameter and how it is displayed to patt	ern users.		
			1
Configure an XPath expression to transform the value of this pa The XPath expression can get the value of other parameters by	arameter. The XPath expres / using the getValue() XPath	sion is evaluated when a pattern instance is cre function.	eated.
Functions		Operators	
रि कि कि and the second se		+	
🛨 📲 Number			
			_
Function name:	뒷 Use	Operator:	뒷 Use
-			
Pattern Parameters			
Groups and Parameters	Parameter ID	Test Value	
E Is Application Queue Information     Oucue pame	queueName	queue	
Queue prefix	queuePrefix	QP.PRODUCTION.	
🖹 Queue suffix	queueSuffix	.qs	
Test value: QP.PRODUCTION.	😤 Set 🛛 Pa	rameter ID: queuePrefix	😽 Use
Expression Evaluation			
Expression: concat(pp:getValue('queuePrefix'),	pp:getValue('queueName'),p	p:getValue('queueSuffix'))	📄 Evaluate
Result: OP.PRODUCTION.queue.gs			
		•	
			OK Cancel

15. The Result field also contains the details of any failure of the XPath expression syntax.

Append an invalid character to the XPath expression, and click Evaluate.

⇒ Edit Parameter: Queue name			<u>×</u>
Configure the pattern parameter			
An XPath expression for this parameter is invalid			
Basic Editor Transform Enable			1
Configure an XPath expression to transform the value of this parar The XPath expression can get the value of other parameters by us	neter. The XPath expression in the getValue() XPath fu	n is evaluated when a pattern instance is create nction.	d.
Functions		Operators	
E → → ta Boolean			
E		-	
E → te Pattern		div	
		=	
			<b>_</b>
Function name:	ç Use	Operator:	뀻 Use
Pattern Parameters			
	1	[	
Groups and Parameters	Parameter ID	Test Value	
	queueName	queue	
Queue prefix	queuePrefix	QP.PRODUCTION.	
	queuesurrix	.qs	
Test value: OR PRODUCTION	🗘 Set 🛛 Dara	meter ID: gueueBrefix	- R lise
JOPPRODUCTION.		Juederrenx	
Expression Evaluation			
Expression: concat(pp:getValue('queuePrefix'),pp:	getValue('queueName'),pp	getValue('queueSu fix'))£££	⊳ Evaluate
Dearth		·	
Result:   Extra illegal tokens: ±±±			

16. Finally, reset the parameter values to generate the full queue name qp.queue.qs, and click OK.

dit Parameter: Queue name			
nfigure the pattern parameter			
Configure the pattern parameter and how it is displ	aved to pattern users.		
asic Editor Transform Enable			
Configure an XPath expression to transform the va	lue of this parameter. The XPath expression is	evaluated when a pattern instance is created	ł.
he XPath expression can get the value of other pa	arameters by using the getValue() XPath funct	ion.	
unctions	Or	perators	
±		*	<b>_</b>
E standard Stand Standard Standard Stan		*	
		div	
E String		=	
		!=	
		<	•
unction name:	रु Use	Operator:	रु Use
Pattern Parameters			
Pattern Parameters Groups and Parameters	Parameter ID	Test Value	
Pattern Parameters Groups and Parameters	Parameter ID	Test Value	
Pattern Parameters Groups and Parameters Groups and Parameters Comparison Queue Information Comparison Queue name	Parameter ID queueName	Test Value queue	
Pattern Parameters Groups and Parameters C Application Queue Information C Queue name Queue prefix	Parameter ID queueName queuePrefix	Test Value queue qp.	
Pattern Parameters Groups and Parameters Description Queue Information Description Queue name Description Descript	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters Groups and Parameters □ I Application Queue Information C Queue name C Queue prefix C Queue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters Groups and Parameters □ □ □ Application Queue Information □ Queue name □ Queue prefix □ Queue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	_
Pattern Parameters Groups and Parameters □ □ Application Queue Information ♀ Queue name ♀ Queue prefix ♀ Queue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters Groups and Parameters  Groups and Parameters Queue Information Queue name Queue prefix Queue suffix Tect value:	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters Groups and Parameters Qroups and Parameters Queue Information Queue name Queue prefix Queue prefix Queue suffix Test value: qp.	Parameter ID queueName queuePrefix queueSuffix Set Parame	Test Value queue qp. .qs ter ID: queuePrefix	Use
Pattern Parameters          Groups and Parameters         Groups and Parameters         Image: Application Queue Information         Image: Queue name         Image: Queue prefix         Image: Queue prefix         Image: Queue prefix         Image: Queue suffix         Image: Test value:         Image: Queue queue         Image: Queue suffix         Image: Queue queue         Image: Queue <td>Parameter ID queueName queuePrefix queueSuffix Set Parame</td> <td>Test Value queue qp. .qs ter ID: queuePrefix</td> <td>Use</td>	Parameter ID queueName queuePrefix queueSuffix Set Parame	Test Value queue qp. .qs ter ID: queuePrefix	Use
Pattern Parameters          Groups and Parameters         Groups and Parameters         Image: Complex and Parameter	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp, ,qs ter ID: queuePrefix	Use
Pattern Parameters Groups and Parameters Queue Information Queue name Queue prefix Queue suffix Test value:  Expression Evaluation Expression: Concat(pp:getValue(queue)	Parameter ID queueName queuePrefix queueSuffix Set Parame ueuePrefix''),pp:getValue('queueName'),pp:getValue	Test Value queue qpqs ter ID: queuePrefix Value('queueSuffix'))	Use
Pattern Parameters Groups and Parameters Queue Information Queue name Queue prefix Queue suffix Test value: qp. Expression Evaluation Expression: Concat(pp:getValue(queue prefix))	Parameter ID queueName queuePrefix queueSuffix Parame ueuePrefix'),pp:getValue('queueName'),pp:getValue('queueName'),pp:getValue	Test Value queue qpqs ter ID: queuePrefix Value('queueSuffix'))	Use
Pattern Parameters Groups and Parameters Queue Information Queue name Queue prefix Queue suffix Test value:  cpp: Expression Evaluation Expression: concat(pp:getValue(queue.qs))	Parameter ID queueName queuePrefix queueSuffix ueuePrefix'),pp:getValue('queueName'),pp:getValue('queueName'),pp:getValue('queueName'),pp:getValue	Test Value queue qpqs ter ID: queuePrefix Value('queueSuffix'))	Use Evaluate
Pattern Parameters Groups and Parameters  Groups and Parameters  Application Queue Information  Queue name Queue prefix Queue suffix  Test value:	Parameter ID queueName queuePrefix queueSuffix ueuePrefix'),pp:getValue('queueName'),pp:get	Test Value queue qpqs ter ID: queuePrefix Value('queueSuffix'))	Use Evaluate
Pattern Parameters          Groups and Parameters         Groups and Parameters         Queue Information         Queue name         Queue prefix         Queue suffix         Test value:         qp.         Expression:         concat(pp:getValue(queue.qs))	Parameter ID queueName queuePrefix queueSuffix Parame PeuePrefix'),pp:getValue('queueName'),pp:get	Test Value queue qpqs ter ID: queuePrefix Value('queueSuffix'))	Use

17. Now regenerate the pattern plug-ins.

Select the "Create Pattern" tab, and click "Create Pattern Plug-ins".

*MyPattern.patter	n X	
进 Create Pat	tern	
Test your pattern by	y configuring your pattern plug-in information, click "Create Pattern Plug-ins", and click "Launch Worl	kbench".
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:	Lom.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	
If you enable this up so that you car pattern, do not se	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	
After you have cr 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.	
To use the patter Message Broker T	n plug-ins, the pattern user must extract the exported archive files into the default WebSphere oolkit plug-ins directory:	

18. Click "Launch Workbench" to start a second instance of the workbench as before. Accept the specified location of the workspace.

This step may take a few minutes.

19. When the new instance has started, select Patterns Explorer, and click the updated pattern (MyPattern).

Click "Create New Instance", and provide a name for the new pattern instance. Click OK.

Wew Pattern Instance	<u>- 🗆 ×</u>
<b>Create New Instance</b> 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: MyXPathExample	
ОК	Cancel

20. When the Pattern Parameter dialogue window is open, expand the group called "Application Queue Information" (or whatever you have named it).

You will see the three parameters that we have just defined, with their default values.

🚦 MyPatternTest - Pattern Configu	ration	🖽 Transform.msgflow	Pattern specification	🔠 MyXPathExam	ple - Pattern Configuration 🛛
🗞 Configure Pattern	Param	neters			
Provide values for pattern paramete	ers. Click th	ne "Generate" button or click	<u>here</u> to generate a pattern inst	ance.	
i Pattern parameters are ready. C	lick the "G	enerate" button to generate	a pattern instance.		
Pattern Parameters				E E Ĕ	Pattern Parameters Details
<ul> <li>Application Queue Inform</li> </ul>	ation				Application Queue Information
Your own description					
Queue name *	queue				
Queue prefix *	qp.				
Queue suffix *	.qs				
·					
					1
Generate					
Specification Configuration					

21. Expand the "Application Queue Information" section in the Pattern Parameter Details pane, on the right. This will show the text descriptions of the parameters we defined.

MyPatternTest - Pattern Configurat	tion (🖼 Transform.msgflow	Pattern specification	🔡 MyXPa	athE×a	mple - Pattern Configuration 🛛 🏹	
🗞 Configure Pattern Pa	arameters					
ovide values for pattern parameters	. Click the "Generate" button or click	here to generate a pattern ins	stance.			
Pattern parameters are ready. Click	k the "Generate" button to generate	a pattern instance.				
Pattern Parameters		E	E 🖻	Patte	ern Parameters Details	
<ul> <li>Application Queue Informati</li> </ul>	ion				Application Queue Informat	ion
Your own description						
Queue name *	queue		_			
Queue prefix *	qp.				Pattern parameter	Description
Queue suffix *	.qs				Queue name	Describe the parameter here
					Queue prefix	The first part of the queue name
					Queue suffix	The last part of the queue name
Generate						
ecification Configuration						

22. Change the name of the queue to something else, and then click Generate.

🔡 MyPatternTest - Pattern Configuration	🖽 Transform.msgflow	Pattern specification	🔠 *My)	XPa	thExa	mple - Pattern Configuration 🕅	
🖏 Configure Pattern Para	ameters						
Provide values for pattern parameters. Cli	ck the "Generate" button or click	<u>here</u> to generate a pattern insta	ance.				
${f i}$ Pattern parameters are ready. Click the	"Generate" button to generate	a pattern instance.					
Pattern Parameters		Ŧ	🗆 🖻	F	Patter	n Parameters Details	
<ul> <li>Application Queue Information</li> </ul>					•	Application Queue Informat	ion
Your own description							
Queue name * MY.	TEST.QUEUE				ſ		
Queue prefix * qp.						Pattern parameter	Description
Queue suffix * .qs						Queue name	Describe the parameter here
						Queue prefix	The first part of the queue name
						Queue suffix	The last part of the queue name
I							
Generate							
Specification Configuration							

23. When the pattern instance has been generated, open the generated message flow.

In this exercise, if you have followed the suggested naming conventions, this will be defined in the MyXPathExample\_Transform project, and will be called Transform.msgflow.

Open the flow, and select the MQInput node. You will see that the queue name has been generated using the default and specific values that you set.



24. We will now make the queue prefix and queue suffix components "read only" in the pattern instance.

Close the second instance of the Message Broker Toolkit, and return to the pattern authoring editor in the primary instance. Select the "Pattern Configuration" tab, which should appear something like this:

🚼 *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	
Image: Constant of the second seco	Add Group Add Parameter Edit Delete Enumerated Types
Java and PHP Code	
	Add Edit Delete
Source Files Pattern Configuration Categories Create Pattern	

25. Select (click on) the "Queue prefix" item, then click the Edit button (or double-click).

dit Parameter: Queue prefix					×
nfigure the pattern parameter					
Configure the pattern parameter and how it is di	splayed to pattern users.				
asic Editor Transform Enable					
Descenter Diseles					
Display name: Queue prefix	Parameter	r ID:	queuePrefix		
Parameter Options					
Hide the parameter	Select this option to hide the parame when a pattern instance is created.	ter and to use an X	Path expression to set the	e value of the parame	iter
Configure during deployment	Select this option if the parameter ma the BAR file.	aps to a target prop	perty and you want the pa	attern user to override	eitin
Mandatory parameter	Select this option if the pattern user a field prompt to guide the pattern u	must enter a value ser.	for the parameter. Manda	atory parameters also	display
Field prompt: Enter your queue pro	efix				
Enter any HTML or text that you want to displ Do not include any <html> or <head> tags be parameter HTML file.</head></html>	ay as help text for this parameter. cause the text is inserted into a	Preview paramete	r help		
The first part of the queue name		The first part	t of the queue name		A
	<b>•</b>				V
				ОК	Cancel

26. Click the Enable tab.

Open the Boolean section in the Functions pane, and double-click "false". This will populate the expression field at the bottom of the editor.

This sets the XPath expression to "false", which tells the pattern to make the parameter uneditable.

Click OK.

Note that the Windows title bar indicates the name of the pattern artifact that is being edited. Any selection in the "Groups and parameters" pane does not affect this edit operation.

ÈEdit Parameter: Queue prefix			X
Configure the pattern parameter			
Configure the pattern parameter and how it is displa	ayed to pattern users.		
Basic Editor Transform Enable			
Configure an XPath expression that contro	ols when this parameter is enabled in the Patterr	n Instance editor.	
If the expression evaluates to true, the p	arameter is enabled, otherwise it is disabled.		
Currenting			
		+	
boolean		*	
true		· div	
		·= ·!=	
		<	<b></b>
Function name: false	Use	Operator:	귯 뒷 Use
- Dathaus Davasataus			
Pattern Parameters			
Pattern Parameters Groups and Parameters	Parameter ID	Test Value	
Pattern Parameters Groups and Parameters  Gro	Parameter ID queueName	Test Value queue	
Pattern Parameters Groups and Parameters ☐ I Application Queue Information ♀ Queue name ♀ Queue prefix ♀ Queue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	_
Pattern Parameters Groups and Parameters □ □ □ Application Queue Information	queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters Groups and Parameters □ Groups and Parameters Queue Information Queue name Queue prefix Queue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters Groups and Parameters □  Queue Information Queue name Queue prefix Queue suffix	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	
Pattern Parameters  Groups and Parameters  Groups and Parameters  Queue Information  Queue name  Queue prefix  Queue suffix  Test value:	Parameter ID queueName queuePrefix queueSuffix Raramete	Test Value queue qp. .qs	Jise
Pattern Parameters	Parameter ID queueName queuePrefix queueSuffix Paramete	Test Value queue qp. .qs	Use
Pattern Parameters         Groups and Parameters         Image: Application Queue Information         Image: Queue prefix         Image: Queue Queue Queue Suffix         Image: Queue Suffix         Test value:         Image: Pattern Queue Prefix         Image: Pattern Queue Suffix         Image: Pattern Queue Prefix         Image: Pattern Queue Suffix         Image: Pattern Queue Suffix         Image: Pattern Queue Pattern Queue Suffix         Image: Pattern Queue Suffix <tr< td=""><td>Parameter ID queueName queuePrefix queueSuffix Raramete</td><td>Test Value queue qp. .qs</td><td>Use Evaluate</td></tr<>	Parameter ID queueName queuePrefix queueSuffix Raramete	Test Value queue qp. .qs	Use Evaluate
Pattern Parameters         Groups and Parameters         Image: Application Queue Information         Image: Queue name         Queue prefix         Queue prefix         Queue suffix         Test value:         Expression Evaluation         Expression:         False()         Result:	Parameter ID queueName queuePrefix queueSuffix	Test Value queue qp. .qs	Use Evaluate
Pattern Parameters         Groups and Parameters         Image: Application Queue Information         Image: Queue prefix         Queue prefix         Image: Queue Queue Suffix         Image: Queue Suffix         Test value:         Expression Evaluation         Expression:         Image: Pattern Parameters         Image: Pattern	Parameter ID queueName queuePrefix queueSuffix <b>et</b> Set Paramete	Test Value queue qp. .qs	Use Evaluate
Pattern Parameters         Groups and Parameters         Queue name         Queue prefix         Queue suffix         Pattern         Test value:         Expression         False()         Result:	Parameter ID queueName queuePrefix queueSuffix Raramete	Test Value queue qp. .qs	Evaluate

- 27. Do the same edit action on the Queue Suffix parameter.
- 28. Create the Patterns Plug-ins as before, then launch the new workbench again. Accept the default workspace.
- 29. In the new workbench, switch to the Patterns Explorer view.



30. Click MyPattern, and Create New Instance. Pprovide a name for the pattern instance, for example MyReadonlyExample.

📴 Broker Developm 🙀 Patterns Explorer 🗴 🦵 🗖	Pattern specification 🕱
🖃 📁 Patterns	View Pattern Specification
🚊 🛗 Application Integration	
🖻 🛗 SAP	View information about the selected pattern and click the "Create New Instance" button or click <u>here</u> to start using a pattern.
MQ one-way (IDoc)	
🛱 🛗 File Processing	
E-	Pattern Specification
MQ one-way	II
🗄 🛗 Message-based Integration	New Pattern Instance
🕂 📴 Message Correlator	03
MQ request-response with persistence	Create New Instance
MQ request-response without persister	The instance name uniquely identifies the pattern instance project that is created when the pattern instance is
🖻 🥮 Message Splitter	saved or generated. The name ideally reflects the business and integration functions that the pattern supports.
MQ one-way (XML)	
🕂 🛗 MyVeryOwnCategory	Pattern instance name: MvReadonIvExample
	PSS
🖻 😕 Service Enablement	e fi
🖻 遭 Service Access	
🔡 MQ one-way	
🖻 🥮 Service Facade	
	OK Cancel
MQ request-response	
🖻 🔑 Service Virtualization	a queue name nom where messages are read. Arter the pattern users have c
🖻 🦉 Service Proxy	users generate a Pattern Instance project, which is a WebSphere Message Bi
Static endpoint	configuration file. This configuration file stores the pattern parameters that t
	creates one or more additional webSphere Message Broker projects that typi
	I WebSphere Message Broker resources that implement the pattern.
	Craste New Instance

31. As before, expand the Application Queue Information section. You will see that the queue prefix and suffix parameters are greyed out, but you can change the value of the queue name parameter.

Click Generate to create the message flow as before.

🗄 Pattern specification 🛛 🔡 *MyReadonlyExample - Pattern Configuration 🛛	
🌯 Configure Pattern Parameters	
Provide values for pattern parameters. Click the "Generate" button or click <u>here</u> to generate a	pattern instance.
${f i}$ Pattern parameters are ready. Click the "Generate" button to generate a pattern instance.	
Pattern Parameters	🕀 📄 🚅 🛛 Pattern Parameters Details
Application Queue Information	Application Queue Information
Your own description	
Queue name * queue	
Queue prefix * gp.	
Oueue suffix *	
Generate	
Specification Configuration	

32. Finally, there are many ways of adding much more sophisticated pattern building tools. As a simple example, you can tick the "Hide parameter" checkbox.

In our example, this will result in the queue suffix parameter being hidden from the pattern user. However, the generated message flow will include the full queue name (including the suffix).

Infigure the pattern parameter         Configure the pattern parameter and how it is         asic       Editor         Transform       Enable         Parameter       Display         Display name:       Queue suffix         Parameter Options       Idide the parameter	s displayed to pattern users.  Parameter ID:  GueueSuffix  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 the pattern parameter and how it is asic Editor Transform Enable Parameter Display Display name: Queue suffix Parameter Options IV Hide the parameter	s displayed to pattern users.  Parameter ID:  QueueSuffix  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.
asic Editor Transform Enable Parameter Display Display name: Queue suffix Parameter Options	Parameter ID: queueSuffix 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.
Parameter Display Display name: Queue suffix Parameter Options I Hide the parameter	Parameter ID: queueSuffix 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.
Display name: Queue suffix Parameter Options	Parameter ID: queueSuffix 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.
Parameter Options	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.
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,
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 queue :	suffix
The last part of the queue name	The last part of the queue name

This concludes the Pattern Authoring XPath lab.