

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.

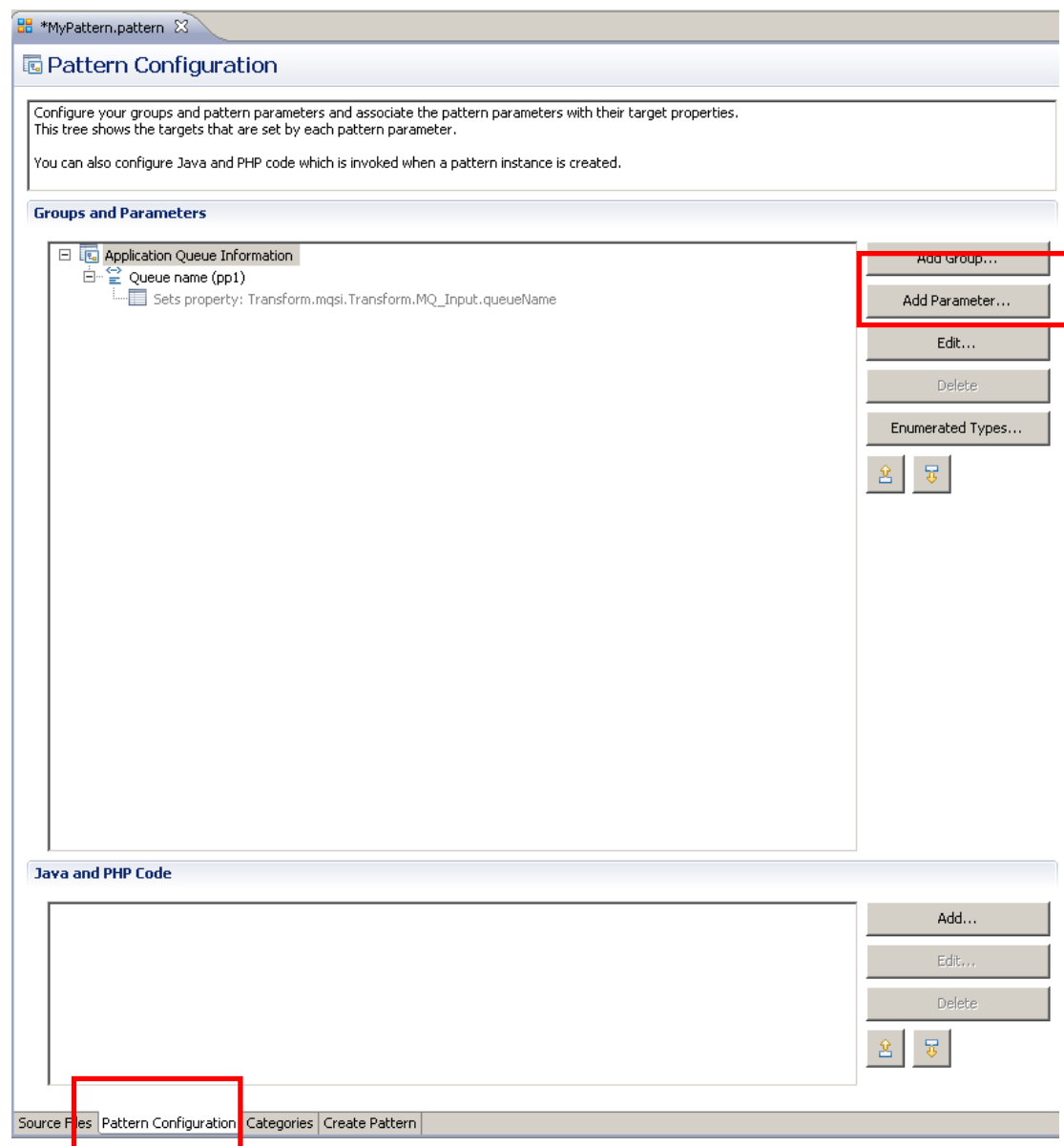
The starting point is the “Pattern Parameters” tab in the Pattern Authoring editor. This lab follows on from the first lab, Pattern Authoring Introduction, which created a pattern with a single parameter, “Queue Name”.

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.

Add Parameter

Configure the pattern parameter

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Parameter Display

Display name: Parameter ID:

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.

☒ 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:

Help Text (HTML)

Enter any HTML or text that you want to display as help text for this parameter. Do not include any <html> or <head> tags because the text is inserted into a parameter HTML file.

Preview parameter help

Describe the parameter here

OK Cancel

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 pattern parameter

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Parameter Display

Display name: Queue prefix Parameter ID: queuePrefix

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.

☒ 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 prefix

Help Text (HTML)

Enter any HTML or text that you want to display as help text for this parameter. Do not include any <html> or <head> tags because the text is inserted into a parameter HTML file.

Preview parameter help

<p>The first part of the queue name</p>

The first part of the queue name

OK Cancel

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).

Click OK.

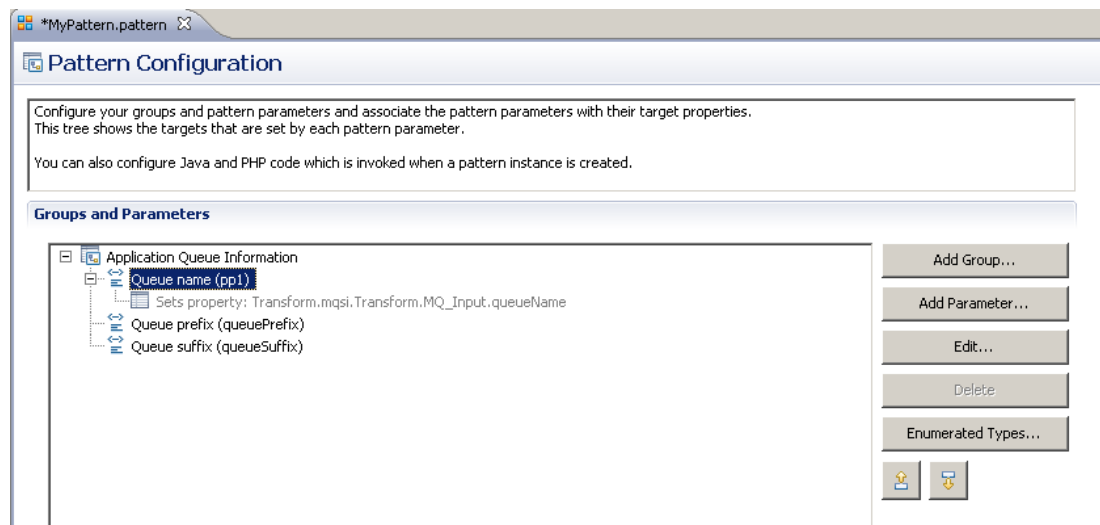
The screenshot shows the 'Add Parameter' dialog box with the 'Editor' tab selected. The dialog has a title bar 'Add Parameter' and a subtitle 'Configure the Pattern Parameter'. Below the subtitle is a description: 'Configure the pattern parameter and how it is displayed to pattern users.' The dialog has four tabs: 'Basic', 'Editor', 'Transform', and 'Enable'. The 'Editor' tab is active. It contains a 'Parameter Editor' section with three fields: 'Parameter editor:' (set to 'Text Editor'), 'Enumerated type:' (empty), and 'Default value:' (set to 'qp.'). There is an 'Enumerated Types...' button next to the 'Enumerated type:' field. Below this is a 'Dependencies' section with explanatory text and a list box for 'This parameter depends on the following parameters:'. At the bottom are 'OK' and 'Cancel' buttons.

5. Perform a similar set of actions for Queue Suffix:

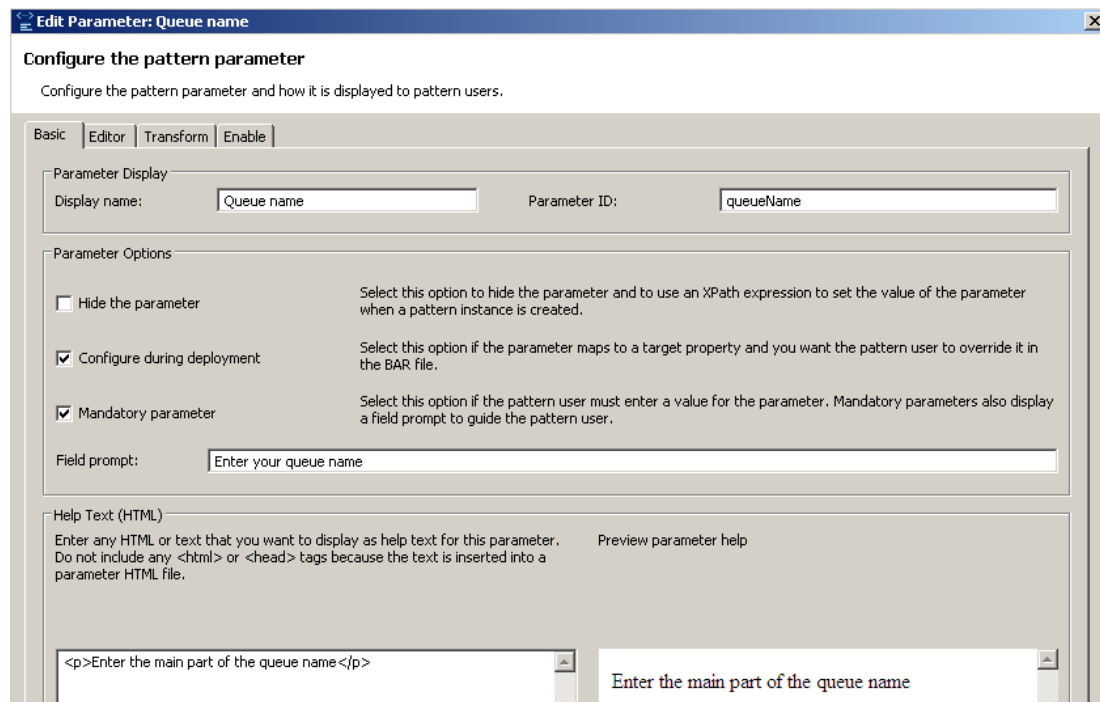
- 1) Click “Add Parameter”
- 2) On the Basic tab:
 - Set Display Name = “Queue suffix”.
 - Set Parameter ID = “queueSuffix”.
 - Set Field Prompt = “Enter your queue suffix”.
 - Set Help text = “The last part of the queue name”.
- 3) On the Editor tab:
 - Set Default value = “.qs”.

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)



Change the Parameter ID to queueName, and click OK.

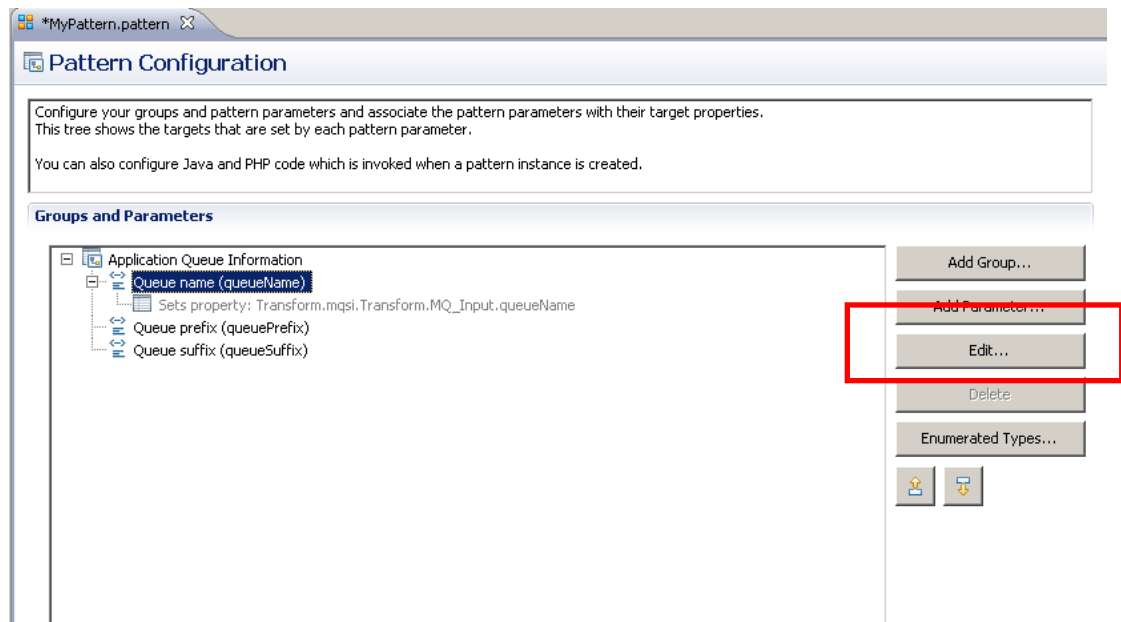


7. You have now defined three pattern parameters in the group Application Queue Information. The parameters are:
- Queue name
 - Queue prefix
 - Queue suffix

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.



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

Edit Parameter: Queue name

Configure the pattern parameter
Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Parameter Display
Display name: Parameter ID:

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.
- ☒ 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:

Help Text (HTML)
Enter any HTML or text that you want to display as help text for this parameter. Do not include any <html> or <head> tags because the text is inserted into a parameter HTML file.

Preview parameter help

OK Cancel

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

Edit Parameter: Queue name

Configure the Pattern Parameter
Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Parameter Editor

Parameter editor:

Enumerated type: Enumerated Types...

Default value:

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 same group and be after the parameter on which it depends.

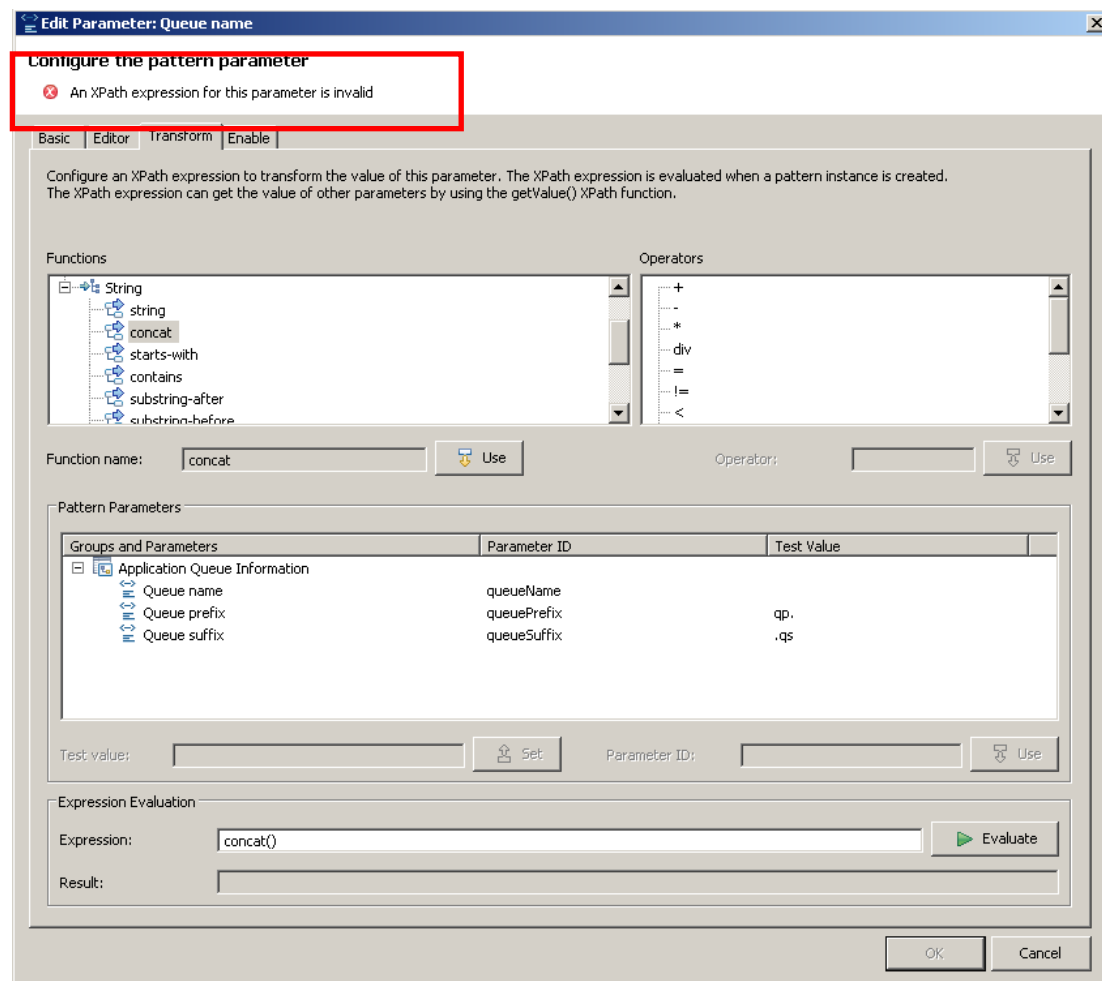
10. Switch to the Transform tab.

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.

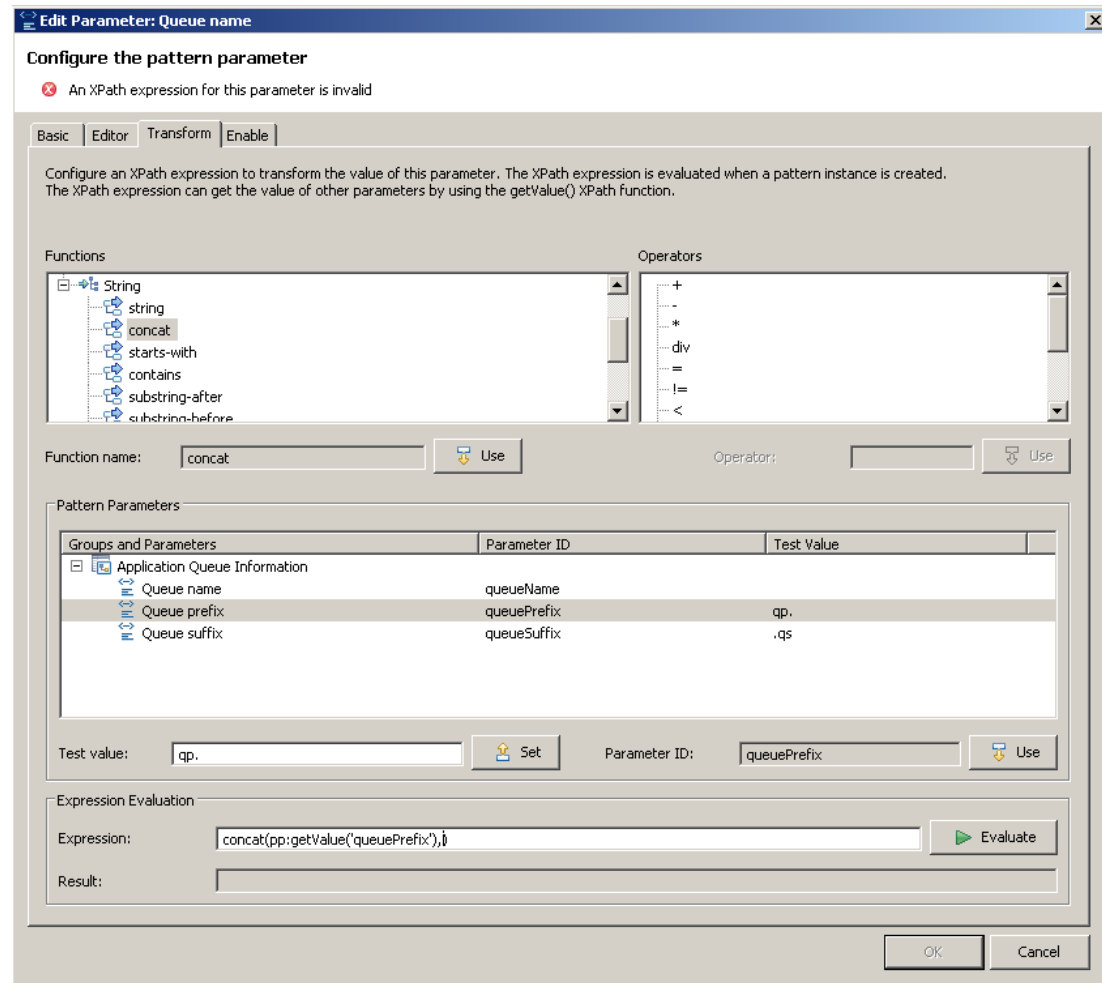
Note that since the expression is incomplete, you will see an error message at the top of the window.



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.

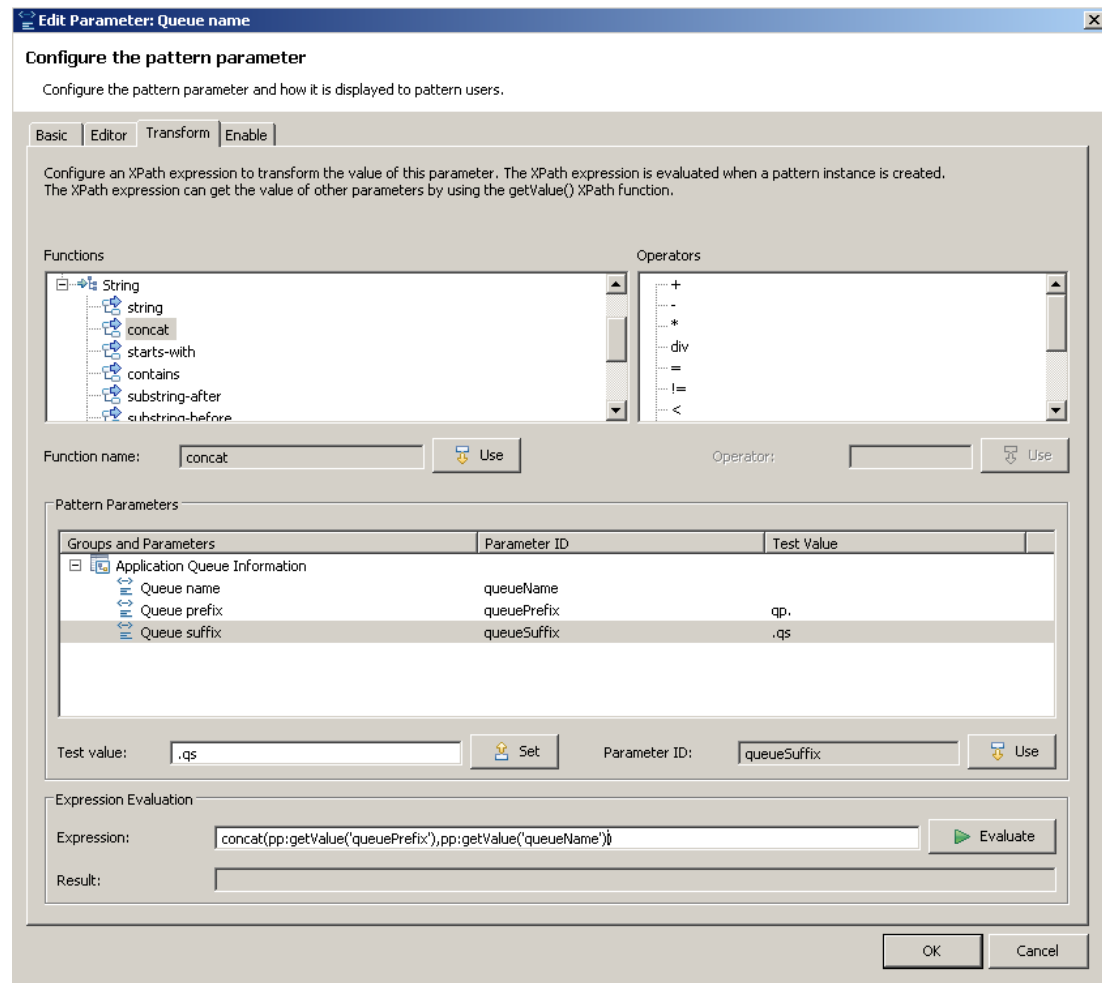
Type a comma “,” after the queue prefix name in the expression field.

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



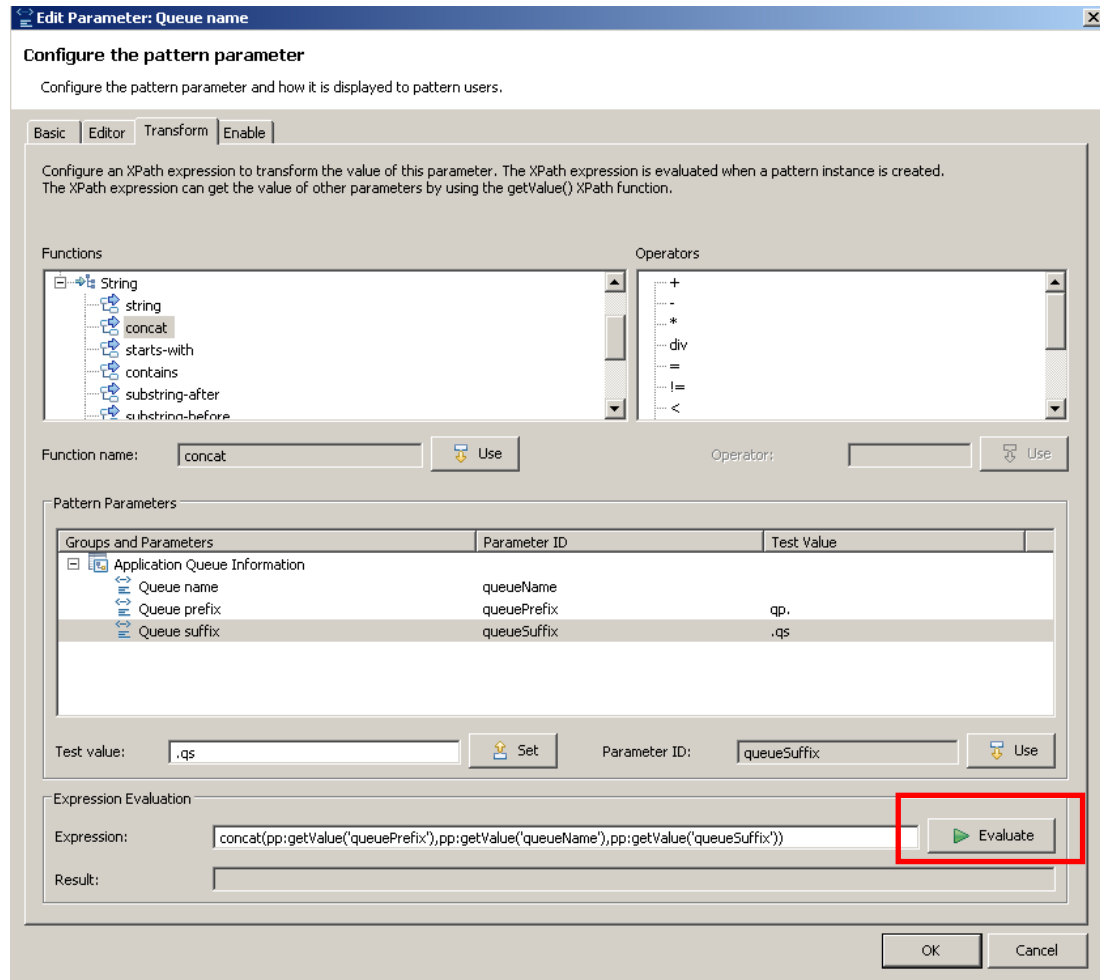
12. Double-click the “Queue name” parameter. This will concatenate the parameter to the XPath expression.

Type a comma “,” as before.



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.



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.

Edit Parameter: Queue name

Configure the pattern parameter

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Configure an XPath expression to transform the value of this parameter. The XPath expression is evaluated when a pattern instance is created. The XPath expression can get the value of other parameters by using the `getValue()` XPath function.

Functions: Boolean, Number, Pattern, String

Operators: +, -, *, div, =, !=, <

Function name: Use Operator: Use

Pattern Parameters

Groups and Parameters	Parameter ID	Test Value
Application Queue Information		
Queue name	queueName	queue
Queue prefix	queuePrefix	QP.PRODUCTION.
Queue suffix	queueSuffix	.qs

Test value: QP.PRODUCTION. Set Parameter ID: queuePrefix Use

Expression Evaluation

Expression: `concat(pp.getValue('queuePrefix'),pp.getValue('queueName'),pp.getValue('queueSuffix'))` Evaluate

Result: QP.PRODUCTION.queue.qs

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

Configure the pattern parameter

✖ An XPath expression for this parameter is invalid

Basic | Editor | **Transform** | Enable

Configure an XPath expression to transform the value of this parameter. The XPath expression is evaluated when a pattern instance is created. The XPath expression can get the value of other parameters by using the `getValue()` XPath function.

Functions

- Boolean
- Number
- Pattern
- String

Operators

- +
-
- *
- div
- =
- !=
- <

Function name: Use

Operator: Use

Pattern Parameters

Groups and Parameters	Parameter ID	Test Value
Application Queue Information		
Queue name	queueName	queue
Queue prefix	queuePrefix	QP.PRODUCTION.
Queue suffix	queueSuffix	.qs

Test value: Set

Parameter ID: Use

Expression Evaluation

Expression: Evaluate

Result:

OK Cancel

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

Edit Parameter: Queue name

Configure the pattern parameter

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | **Transform** | Enable

Configure an XPath expression to transform the value of this parameter. The XPath expression is evaluated when a pattern instance is created. The XPath expression can get the value of other parameters by using the `getValue()` XPath function.

Functions

- Boolean
- Number
- Pattern
- String

Operators

- +
-
- *
- div
- =
- !=
- <

Function name: Use

Operator: Use

Pattern Parameters

Groups and Parameters	Parameter ID	Test Value
Application Queue Information		
Queue name	queueName	queue
Queue prefix	queuePrefix	qp.
Queue suffix	queueSuffix	.qs

Test value: Set

Parameter ID: Use

Expression Evaluation

Expression: Evaluate

Result:

OK Cancel

17. Now regenerate the pattern plug-ins.

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

*MyPattern.pattern

Create Pattern

Test your pattern by configuring your pattern plug-in information, click "Create Pattern Plug-ins", and click "Launch Workbench".

Plug-in Information

Configure the unique identifier for your pattern plug-in.

Pattern name:

Plug-in ID:

Version:

Provider:

Description:

Translation Options

If you enable this option, the Pattern Authoring editor creates two additional NLS plug-ins. These plug-ins are set up so that you can drop in translated resources, such as Java property files. If you are creating a single language pattern, do not select this check box.

☐ Create translation plug-ins (*.nl1 and *.doc.nl1)

Pattern Distribution

After you have created and tested your pattern plug-ins (see the Plug-in ID above for the plug-in names), package your pattern by clicking "File > Export > General > Archive File" to export these plug-ins.

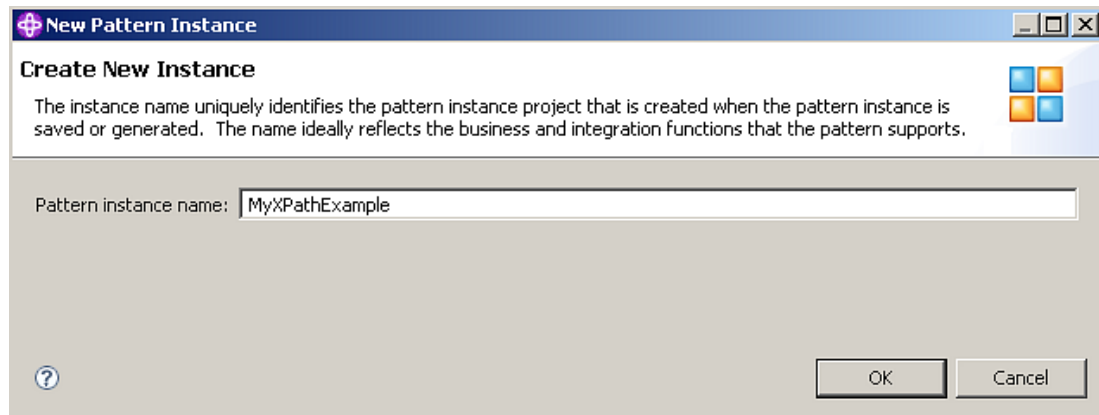
To use the pattern plug-ins, the pattern user must extract the exported archive files into the default WebSphere Message Broker Toolkit plug-ins directory:

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.

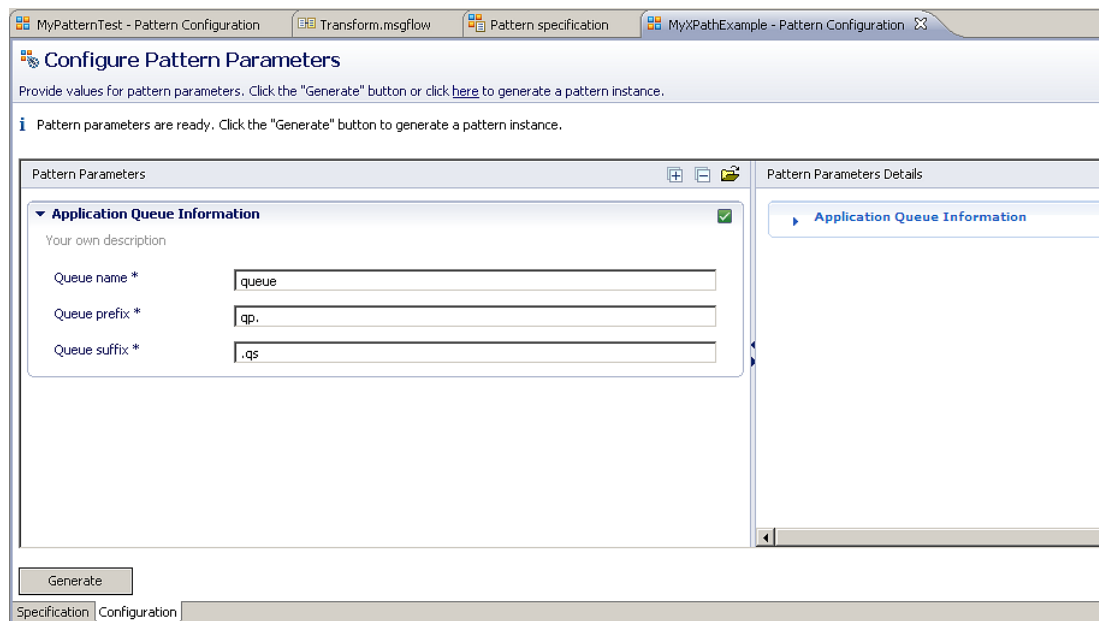
19. When the new instance has started, select Patterns Explorer, and click the updated pattern (MyPattern).

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



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.



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.

Configure Pattern Parameters

Provide values for pattern parameters. Click the "Generate" button or click [here](#) to generate a pattern instance.

Pattern parameters are ready. Click the "Generate" button to generate a pattern instance.

Pattern Parameters

Application Queue Information

Your own description

Queue name *

Queue prefix *

Queue suffix *

Pattern Parameters Details

Application Queue Information

Pattern parameter	Description
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

Specification Configuration

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

Configure Pattern Parameters

Provide values for pattern parameters. Click the "Generate" button or click [here](#) to generate a pattern instance.

Pattern parameters are ready. Click the "Generate" button to generate a pattern instance.

Pattern Parameters

Application Queue Information

Your own description

Queue name *

Queue prefix *

Queue suffix *

Pattern Parameters Details

Application Queue Information

Pattern parameter	Description
Queue name	Describe the parameter here
Queue prefix	The first part of the queue name
Queue suffix	The last part of the queue name

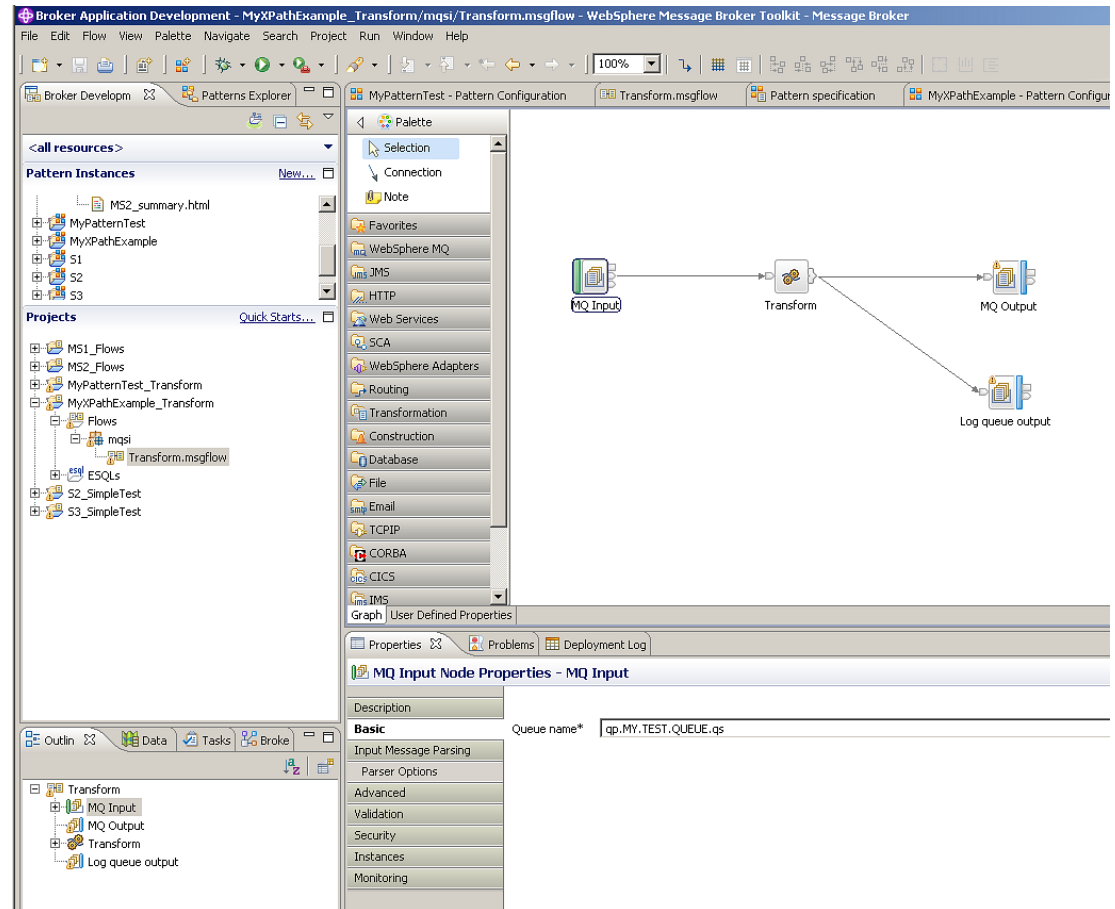
Generate

Specification Configuration

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

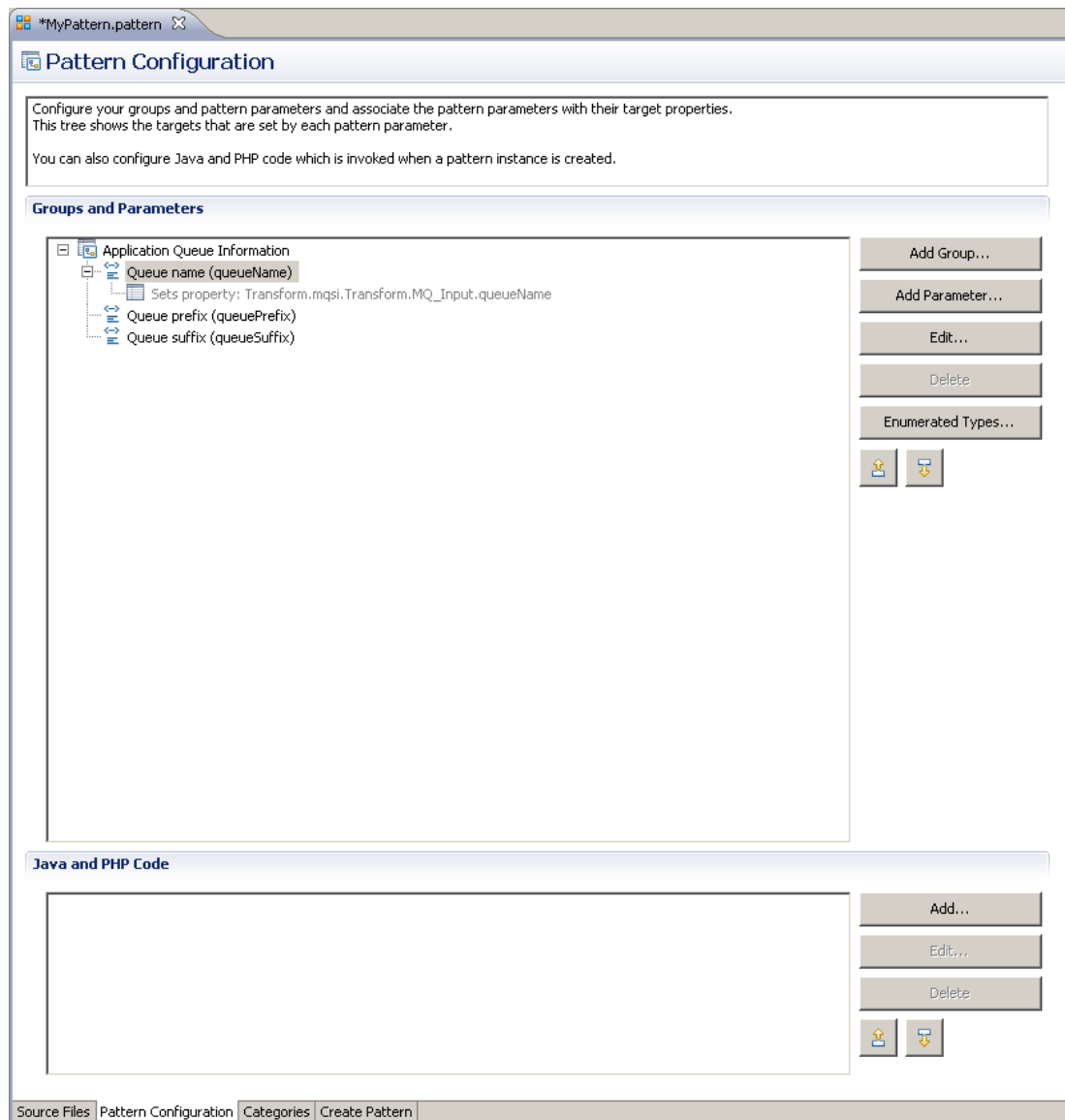
In this exercise, if you have followed the suggested naming conventions, this will be defined in the MyXPathExample_Transform project, and will be called Transform.msgflow.

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:



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

Edit Parameter: Queue prefix

Configure the pattern parameter

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Parameter Display

Display name: Parameter ID:

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.

☒ 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:

Help Text (HTML)

Enter any HTML or text that you want to display as help text for this parameter. Do not include any <html> or <head> tags because the text is inserted into a parameter HTML file.

Preview parameter help

The first part of the queue name

OK Cancel

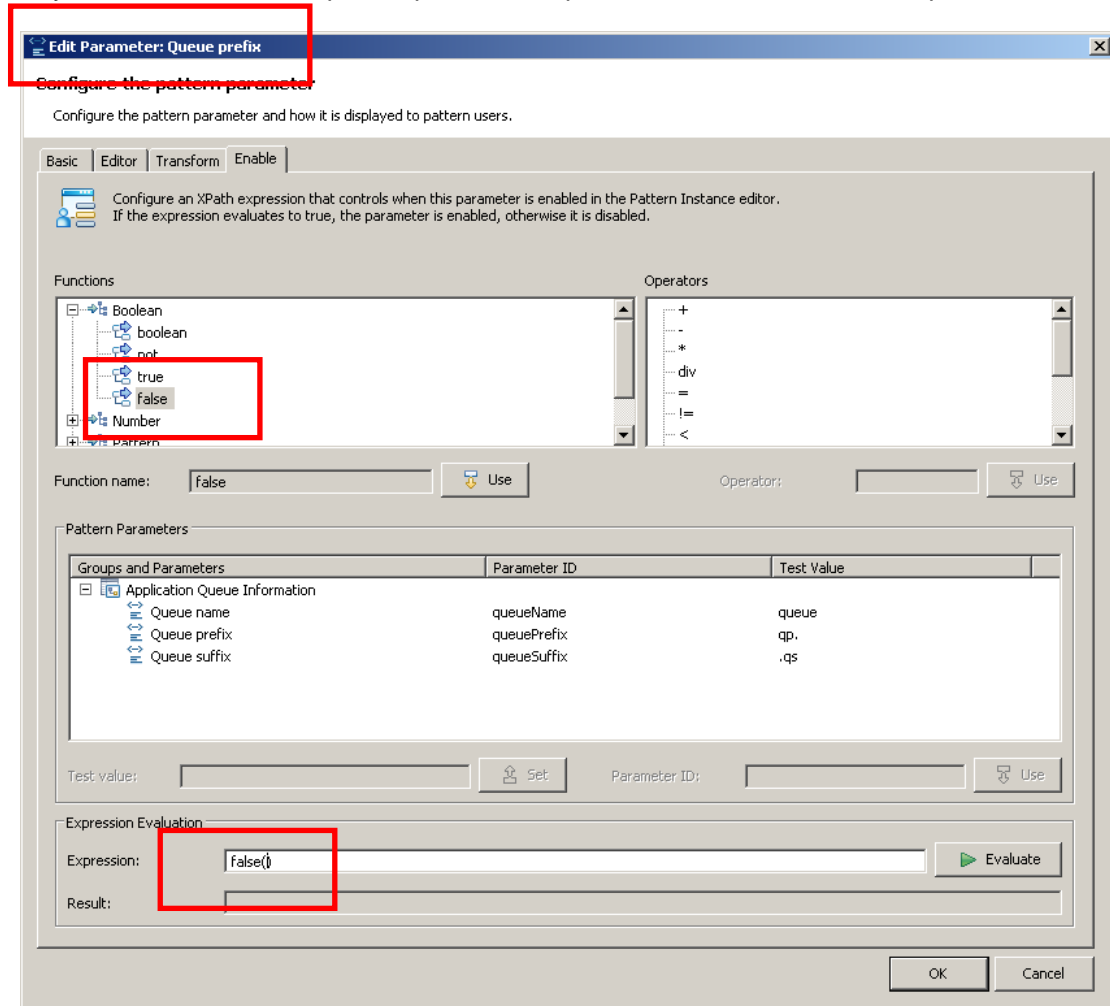
26. Click the Enable tab.

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

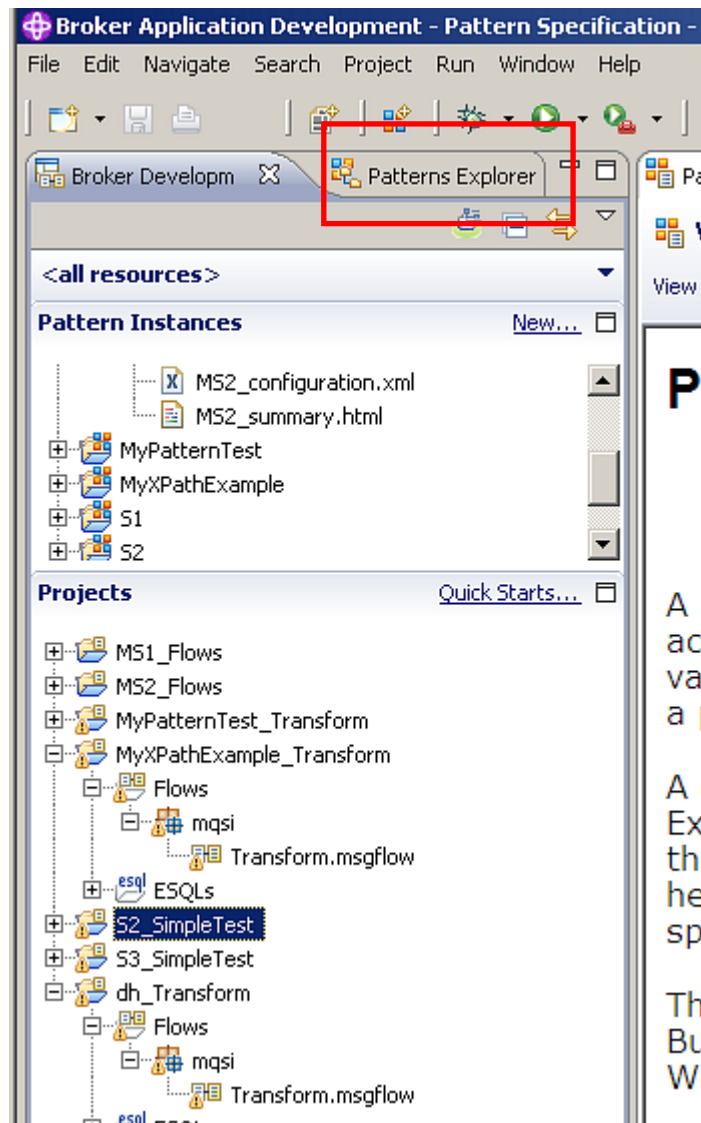
This sets the XPath expression to “false”, which tells the pattern to make the parameter un-editable.

Click OK.

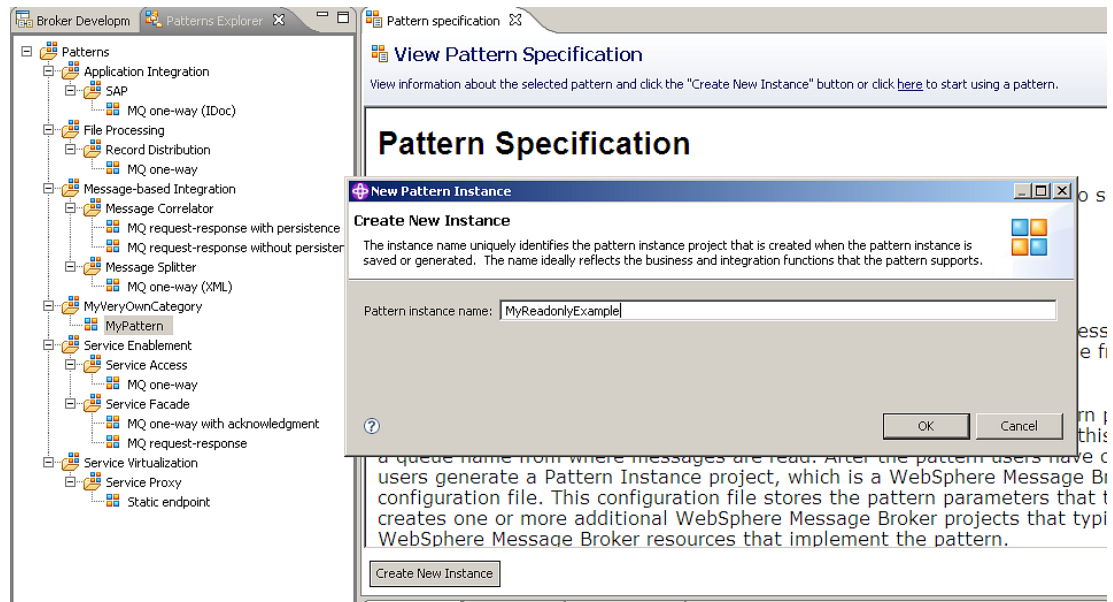
Note that the Windows title bar indicates the name of the pattern artifact that is being edited. Any selection in the “Groups and parameters” pane does not affect this edit operation.



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.

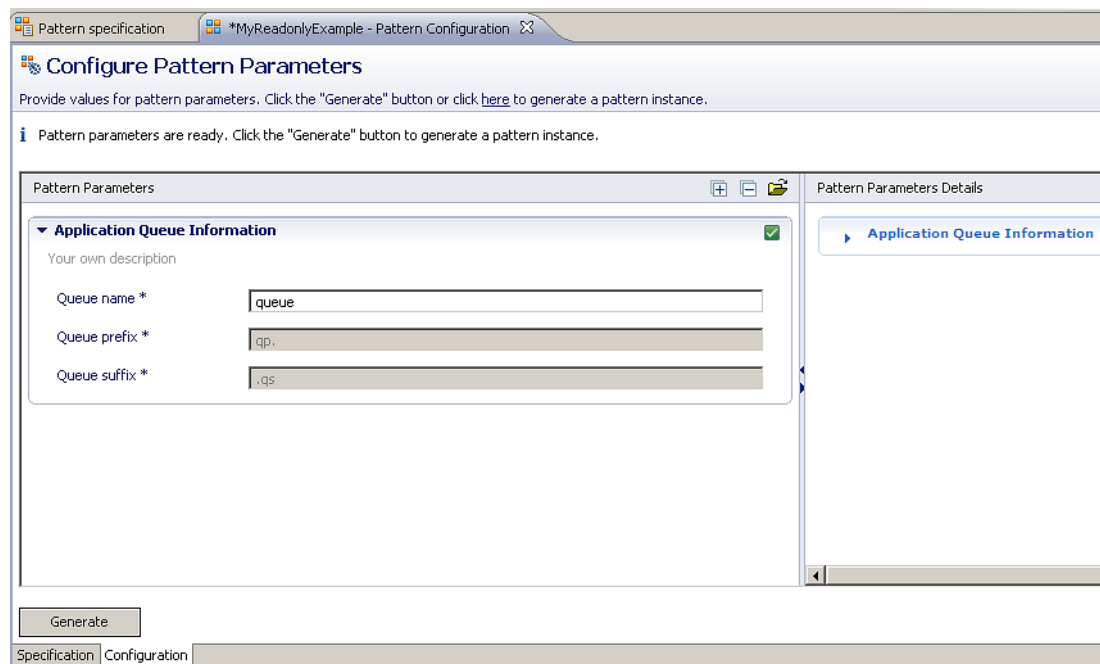


30. Click MyPattern, and Create New Instance. Provide a name for the pattern instance, for example MyReadOnlyExample.



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.



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).

Edit Parameter: Queue suffix

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Editor | Transform | Enable

Parameter Display

Display name: Queue suffix Parameter ID: queueSuffix

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.
- ☒ 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

Help Text (HTML)

Enter any HTML or text that you want to display as help text for this parameter. Do not include any <html> or <head> tags because the text is inserted into a parameter HTML file.

Preview parameter help

<p>The last part of the queue name</p>

The last part of the queue name

OK Cancel

This concludes the Pattern Authoring XPath lab.