

IBM Integration Bus

Pattern Authoring Lab 2

Extending Pattern Authoring with XPath

June, 2013

Hands-on lab built at product
code level version 9.0.0.0

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.

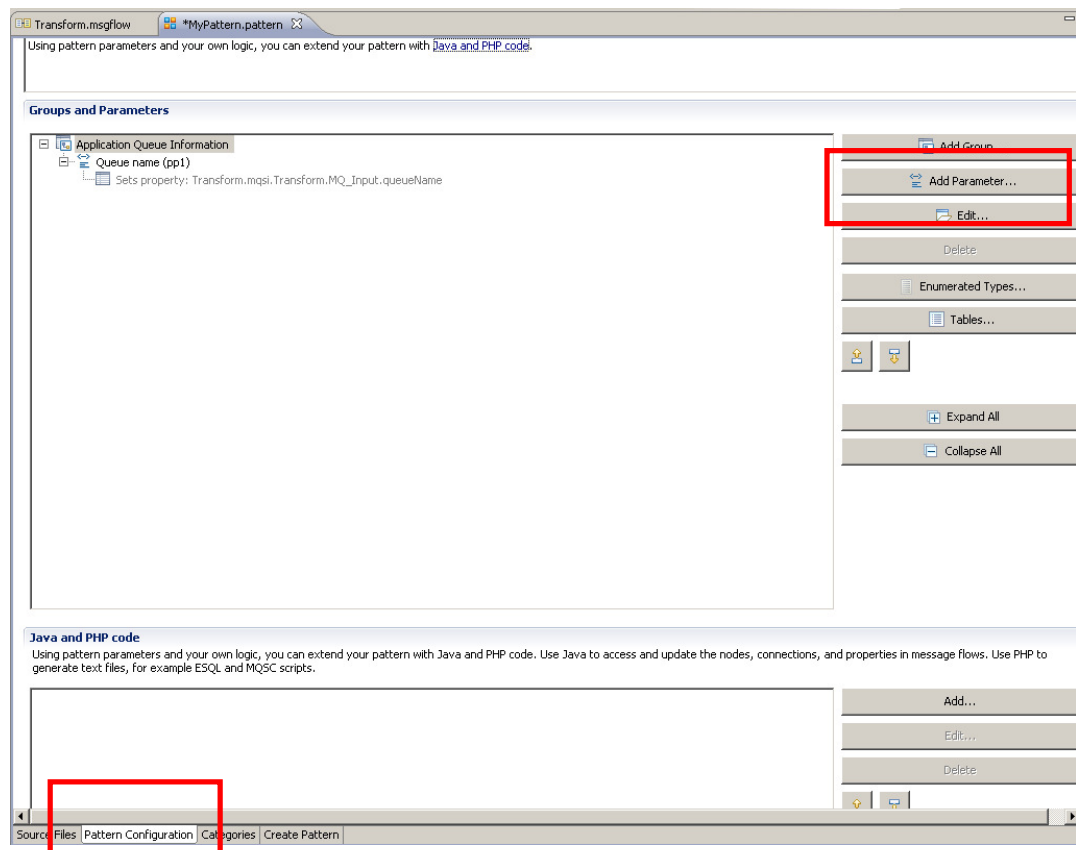
The starting point is the “Pattern Parameters” tab in the Pattern Authoring editor. This lab follows on from the first lab, Pattern Authoring Introduction, which created a pattern with a single parameter, “Queue Name”.

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 IBM Integration Bus 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 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.

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

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 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.

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 cueue name</p>

The first part of the queue name

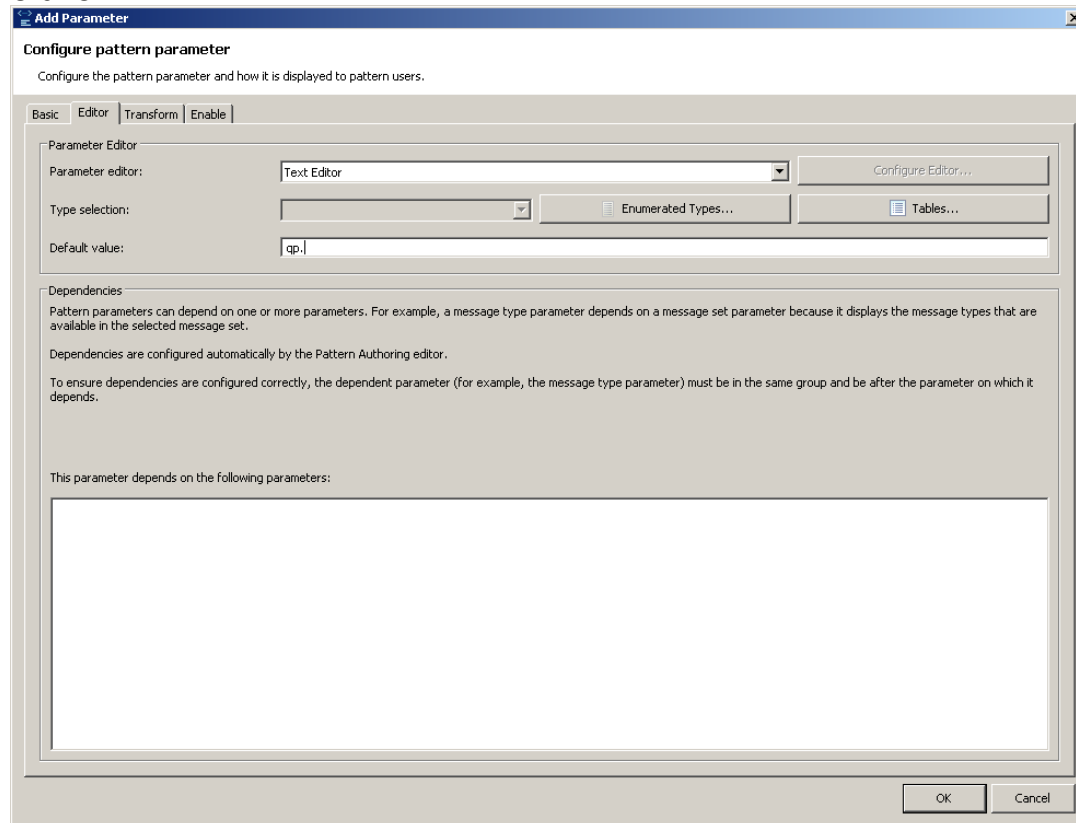
OK Cancel

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.

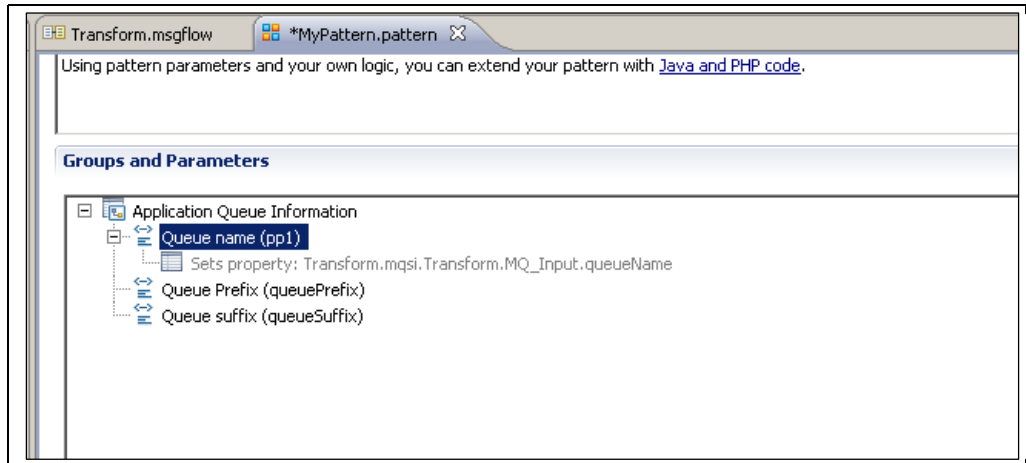


5. Perform a similar set of actions for Queue Suffix:

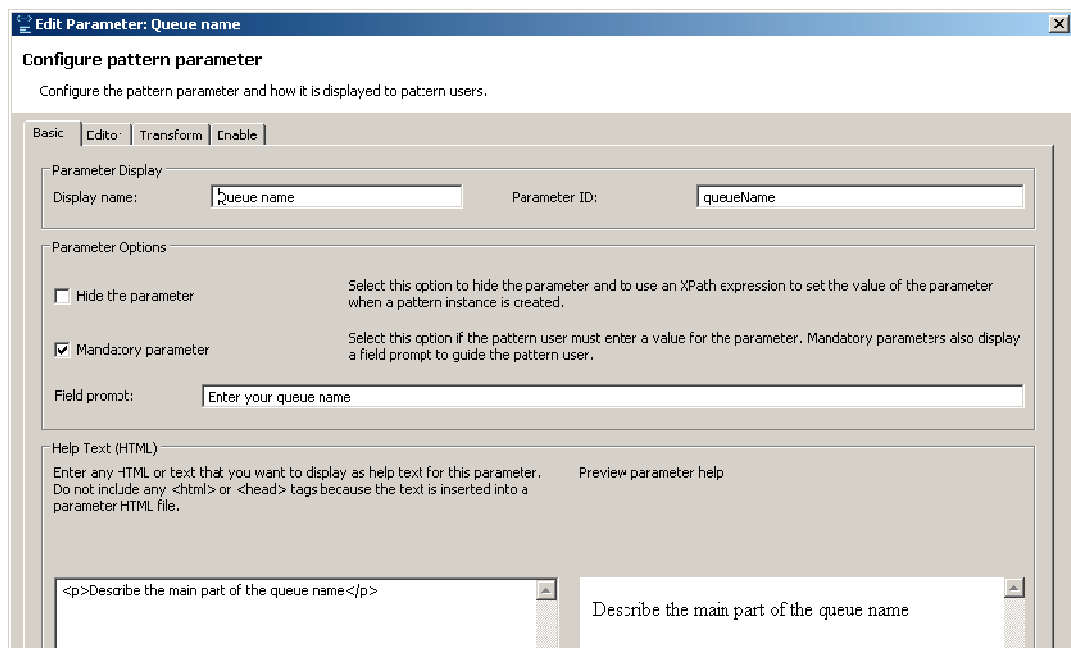
- 1) Click “Add Parameter”
- 2) On the Basic tab:
 - Set Display Name = “Queue suffix”.
 - Set Parameter ID = “queueSuffix”.
 - Set Field Prompt = “Enter your queue suffix”.
 - Set Help text = “The last part of the queue name”.
- 3) On the Editor tab:
 - Set Default value = “.qs”.

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.

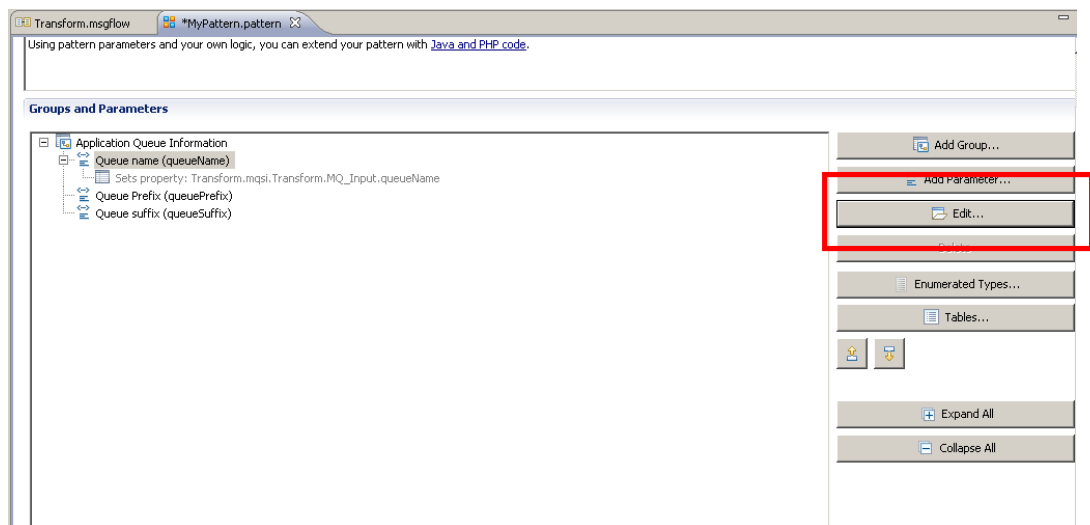


7. You have now defined three pattern parameters in the group Application Queue Information. The parameters are:
- Queue name
 - Queue prefix
 - Queue suffix

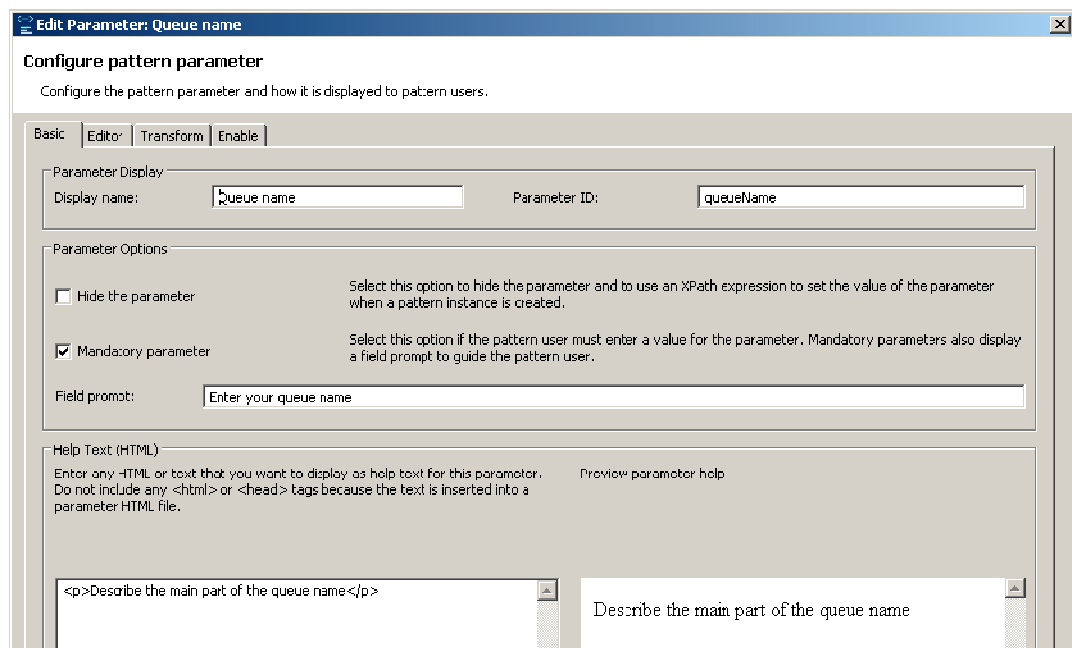
You have also provided meaningful names for the Parameter IDs.

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.



9. On the Editor tab, set the Parameter editor from the drop down menu to “Queue Name Editor” specify the default value for the Queue Name as “queue”.

Edit Parameter: Queue name

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

Basic | **Editor** | Transform | Enable

Parameter Editor

Parameter editor: Queue Name Editor Configure Editor...

Type selection: Enumerated Types... Tables...

Default value: queue

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.

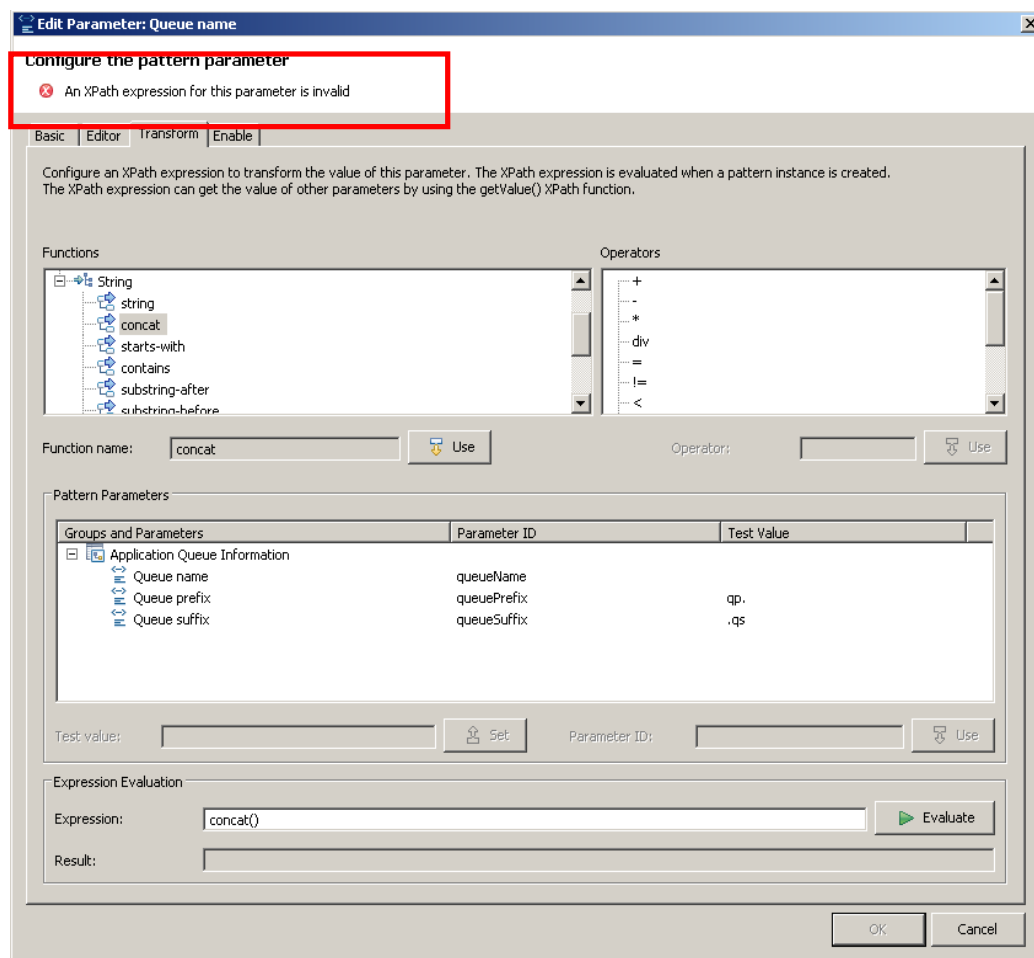
10. Switch to the Transform tab.

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.

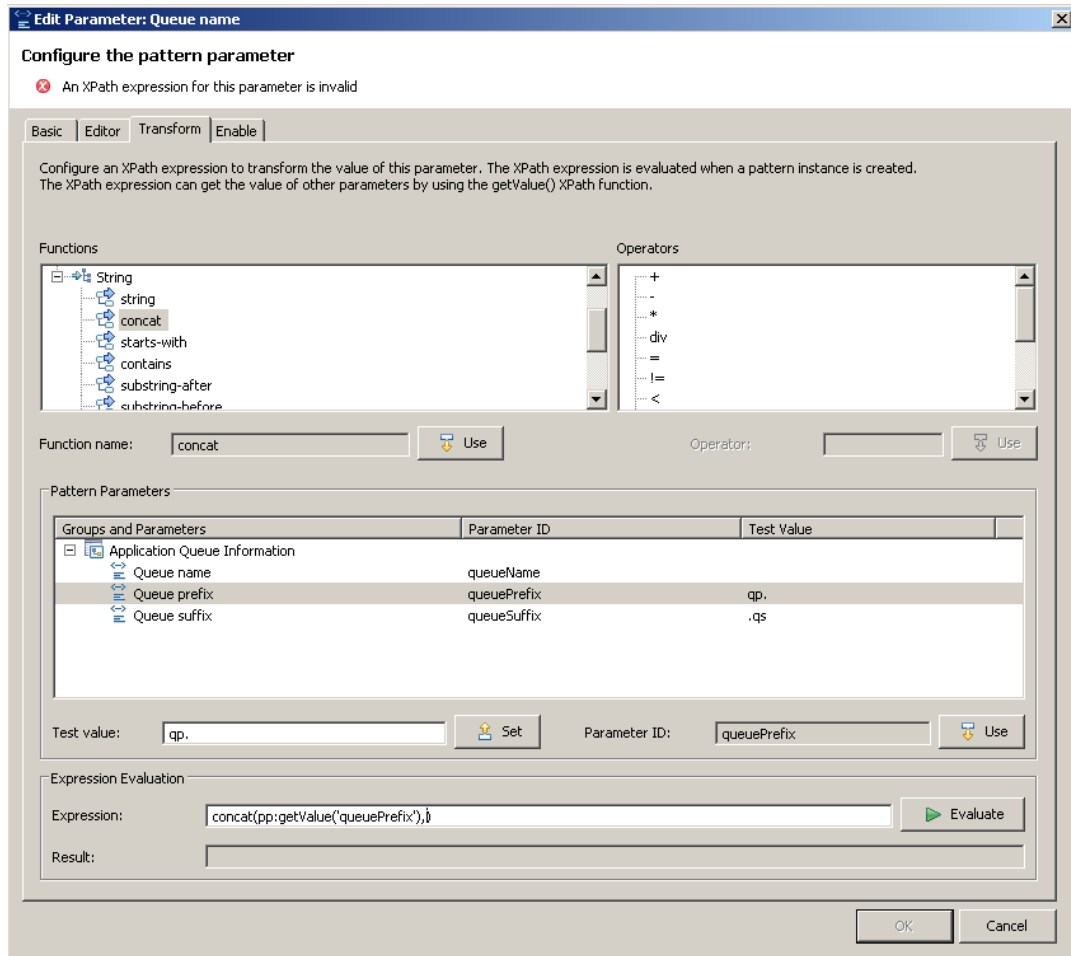
Note that since the expression is incomplete, you will see an error message at the top of the window.



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.

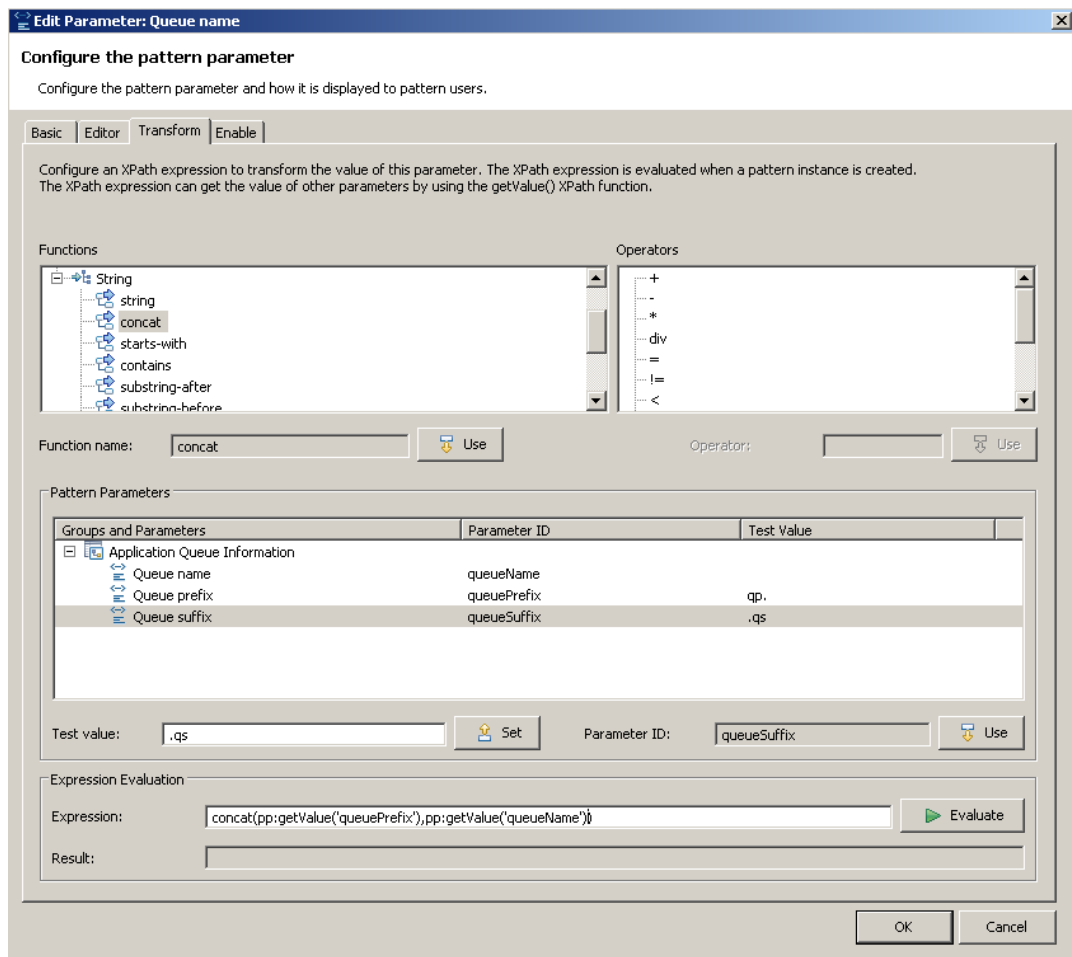
Type a comma “,” after the queue prefix name in the expression field.

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



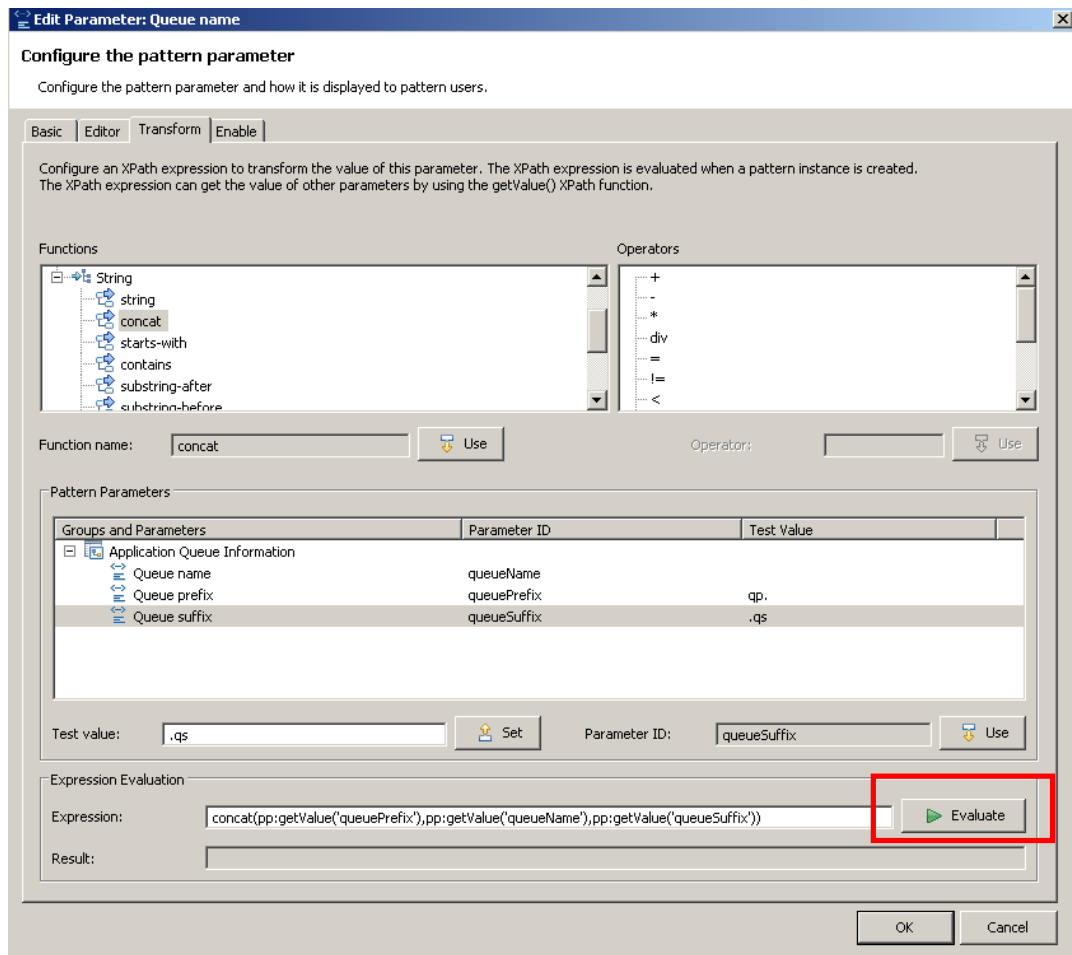
12. Double-click the “Queue name” parameter. This will concatenate the parameter to the XPath expression.

Type a comma “,” as before.



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

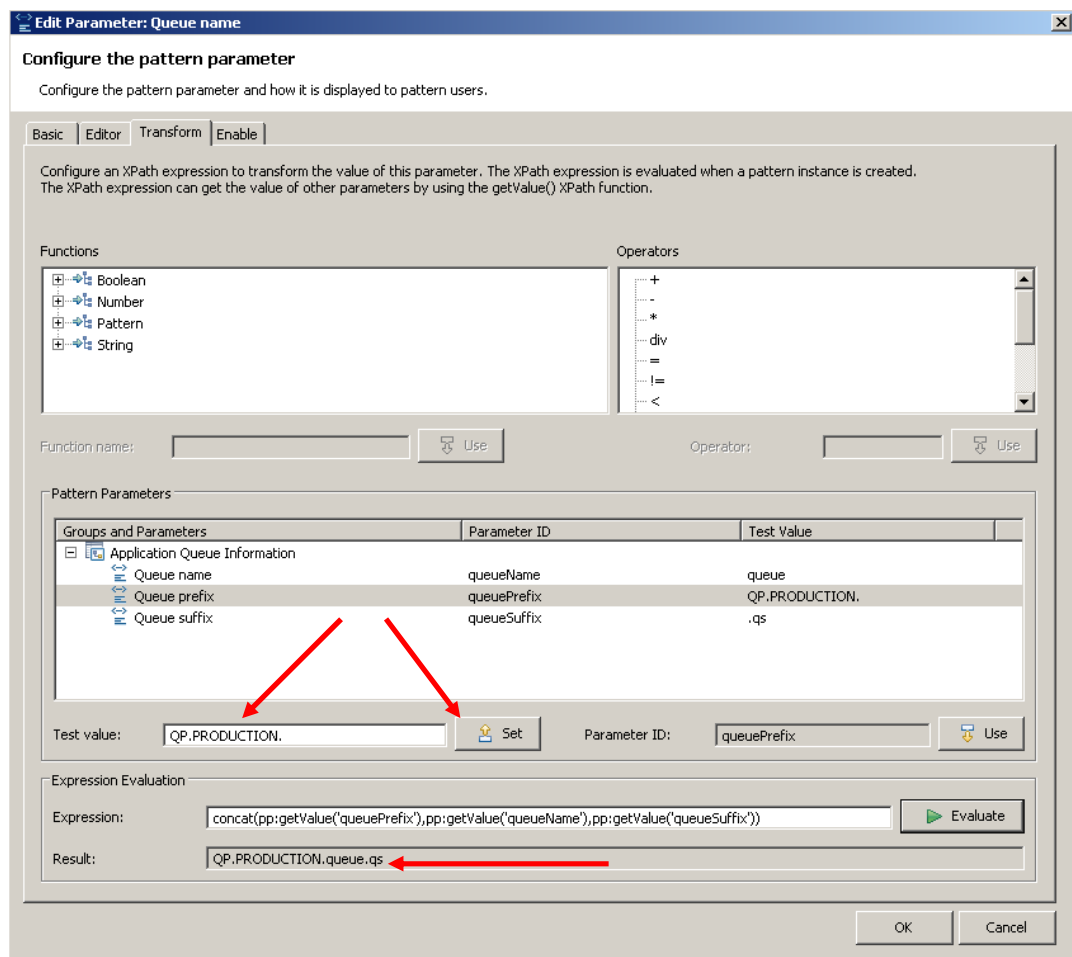
Click the “Evaluate” button to make sure that the XPath expression logic is correct. The Evaluate function uses the default values for each pattern parameter.



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

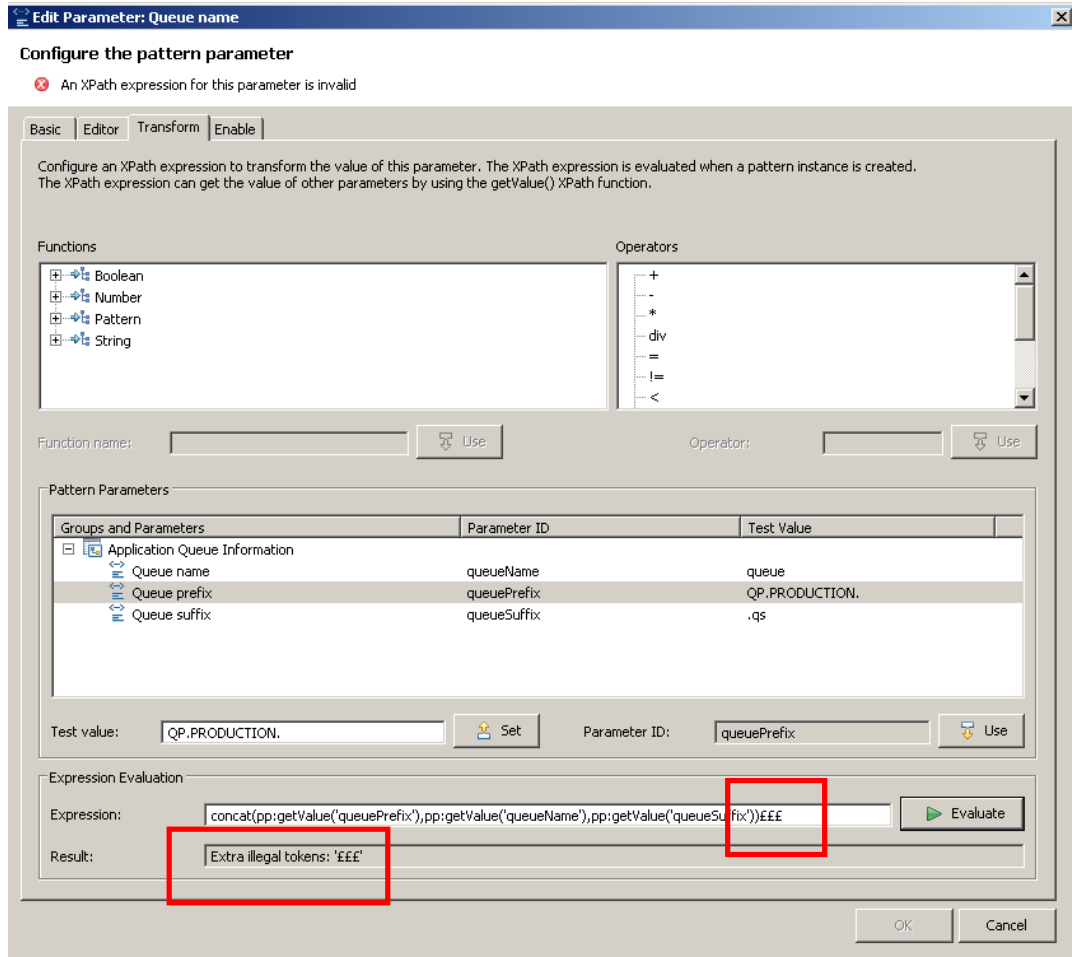
Highlight the “Queue prefix” parameter, and set the Test value field to “QP.PRODUCTION.”. Click Set.

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



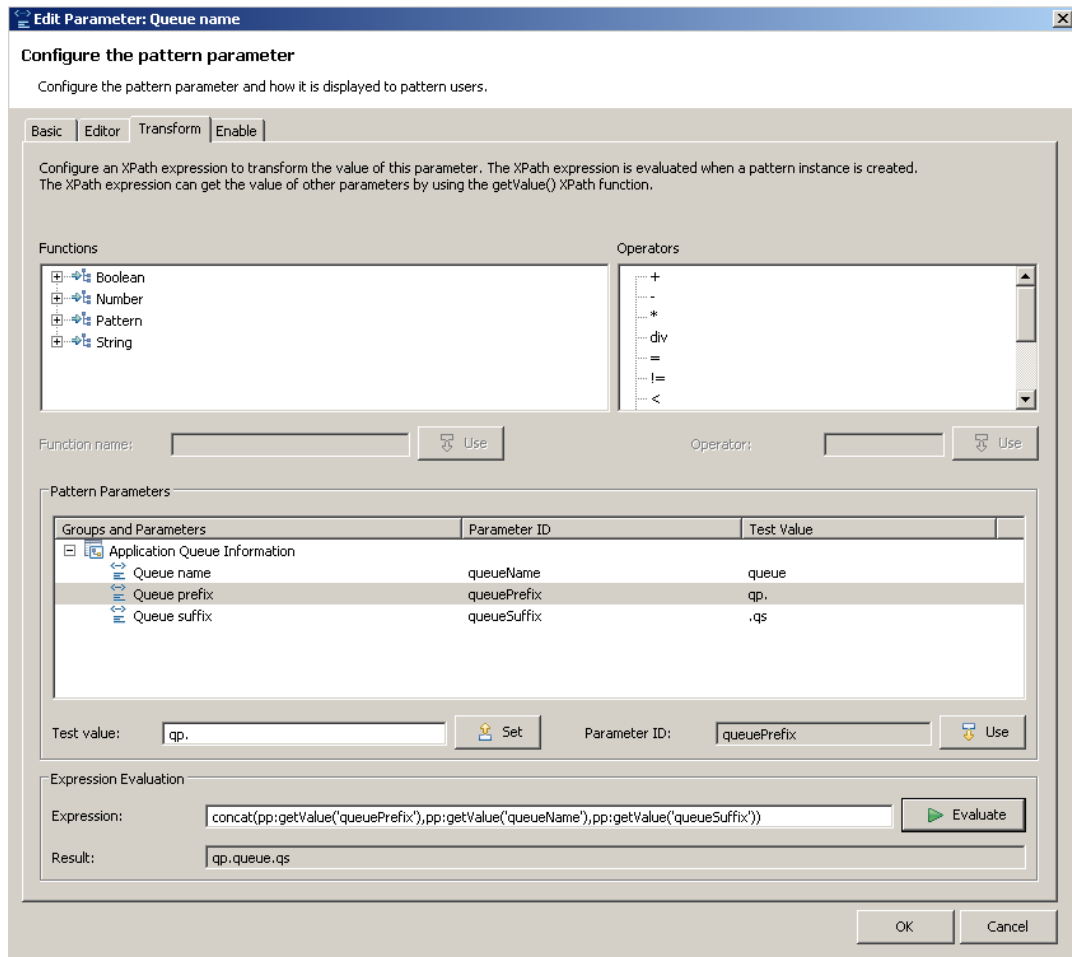
- The Result field also contains the details of any failure of the XPath expression syntax.

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



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

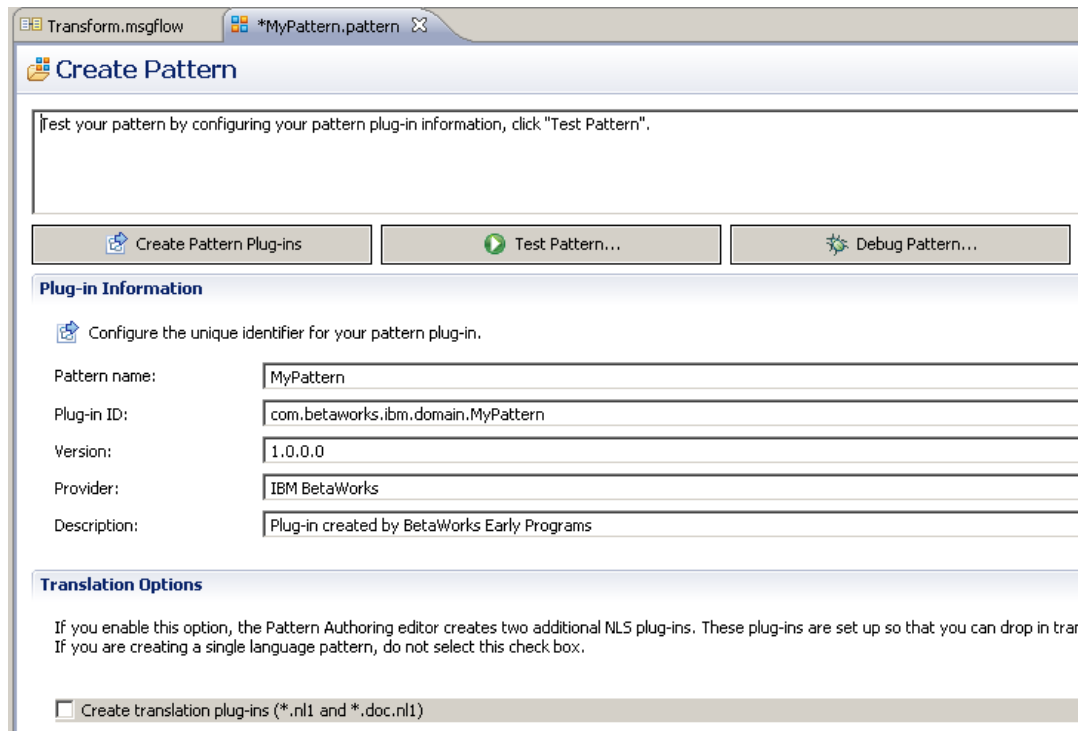
Save MyPattern.pattern.



17. Now retest the pattern.

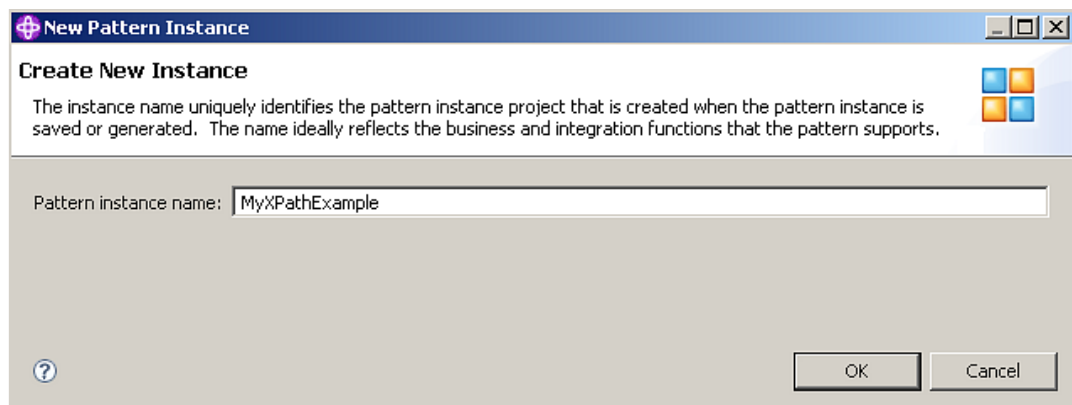
Select the “Create Pattern” tab. The pattern plug-ins will need to be updated, so click “Create Pattern Plug-ins”.

Then click “Test Pattern”. As before, this will start a second instance of the workbench as before. Accept the specified location of the workspace.



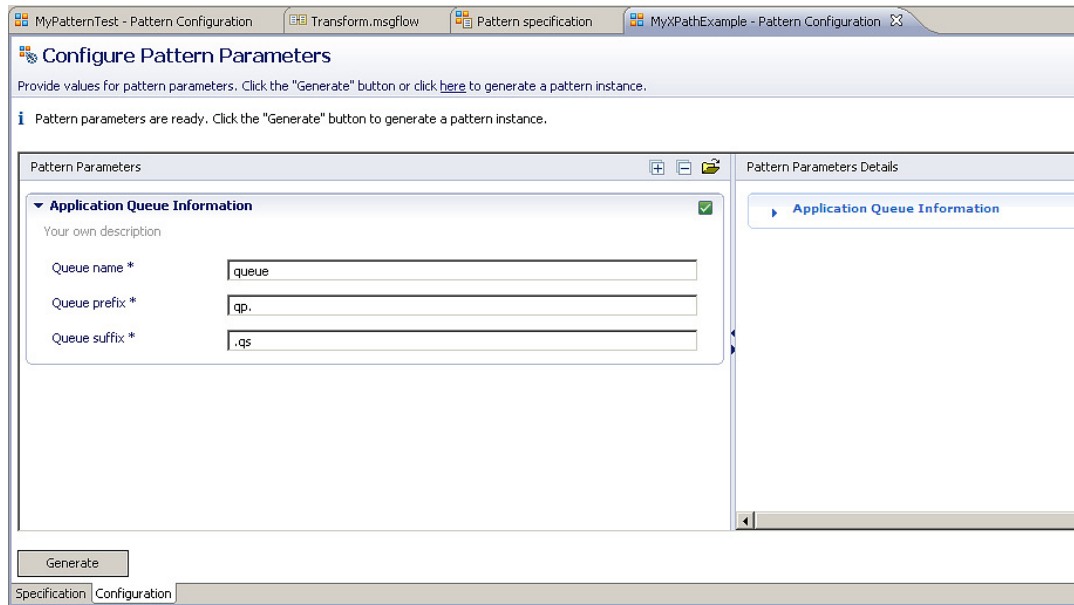
18. When the new instance has started, select Patterns Explorer, and click the updated pattern (MyPattern).

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

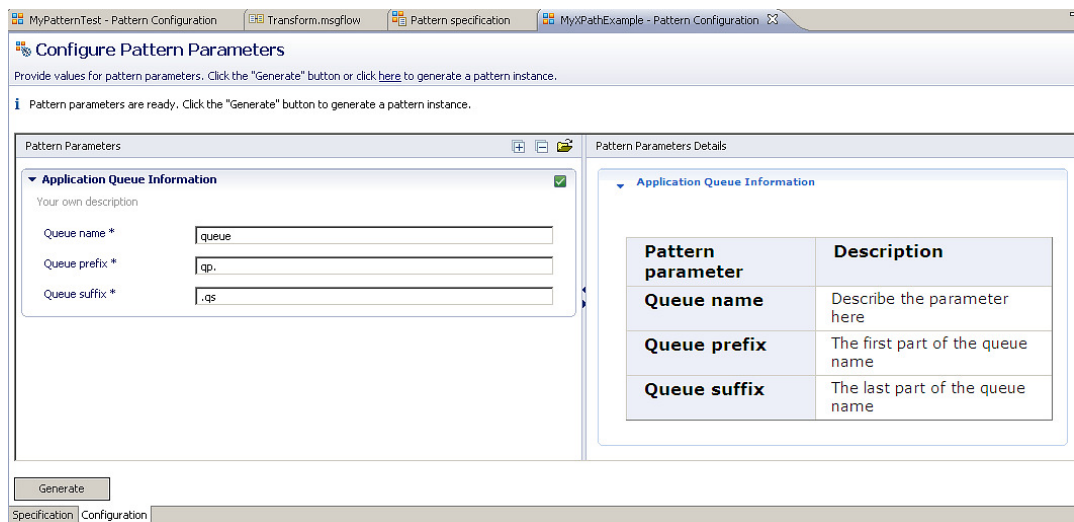


- When the Pattern Parameter dialogue window is open, expand the group called “Application Queue Information” (or whatever you have named it).

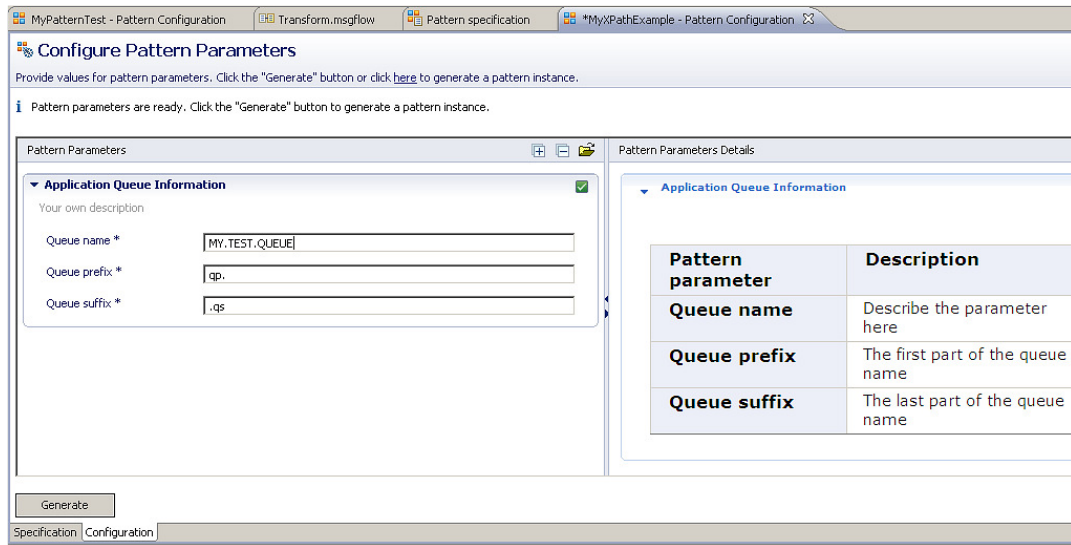
You will see the three parameters that we have just defined, with their default values.



- 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.

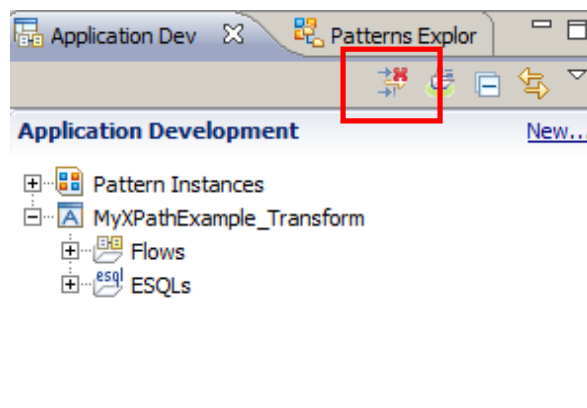


21. Change the name of the queue to something else, and then click Generate.



22. Because the previous pattern generation created a navigator filter, you may find that the newly generated Application does not immediately show in the navigator.

Click the "Remove All Filters" icon (with the red cross).



23. Expand the MyXPathExample_Transform application, and open the generated message flow, which should be called Transform.msgflow.

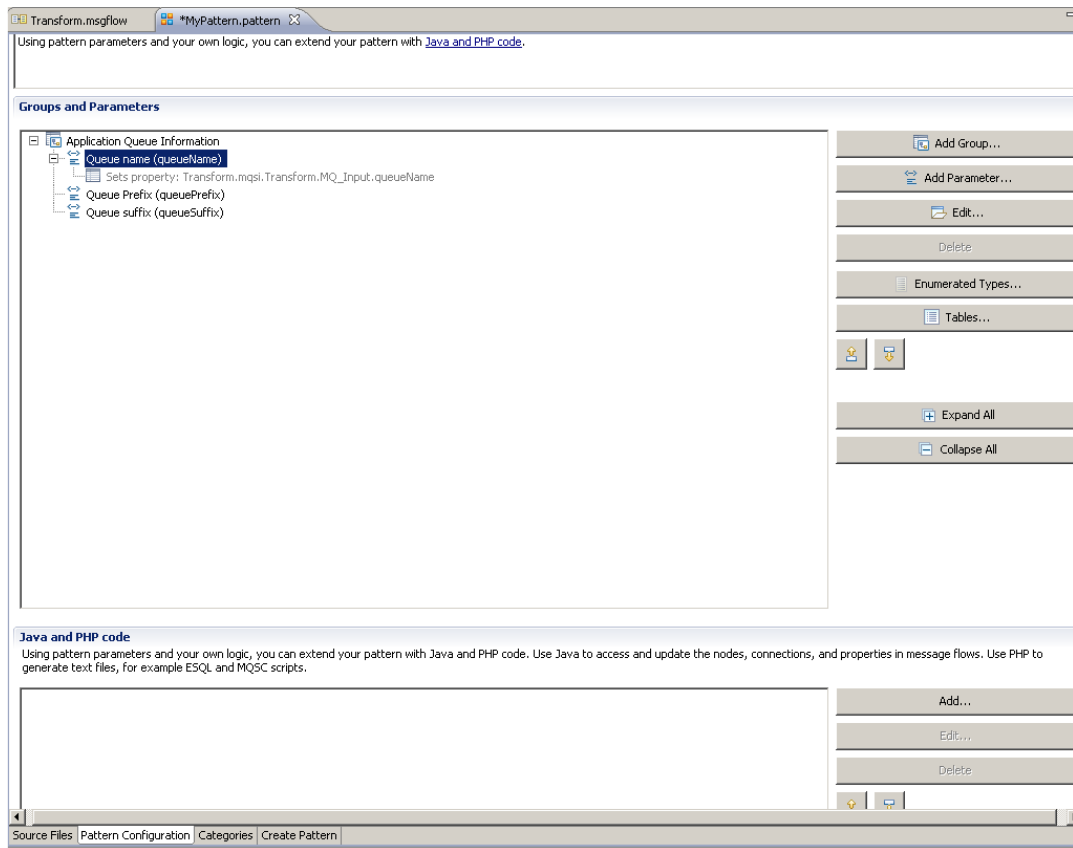
Select the MQInput node. You will see that the queue name has been generated using the default and specific values that you set.

The screenshot displays the IBM Integration Toolkit interface for the 'MyXPathExample_Transform' application. The main workspace shows a message flow diagram with three nodes: 'MQ Input', 'Transform', and 'MQ Output'. A fourth node, 'Log queue output', is also present. The 'MQ Input Node Properties' dialog is open, showing the 'Queue name*' field set to 'qp.MY.TEST.QUEUE.qs'. The 'Integration Nodes' pane at the bottom left shows the application structure, including 'IB9NODE' and 'IB9NODE2'. The 'Properties' pane at the bottom right shows the 'MQ Input Node Properties' dialog with the following fields:

- Description
- Basic: Queue name*
- Input Message Parsing
- Parser Options

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

Close the second instance of the IBM Integration Bus 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).

Add Parameter

Configure 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.

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

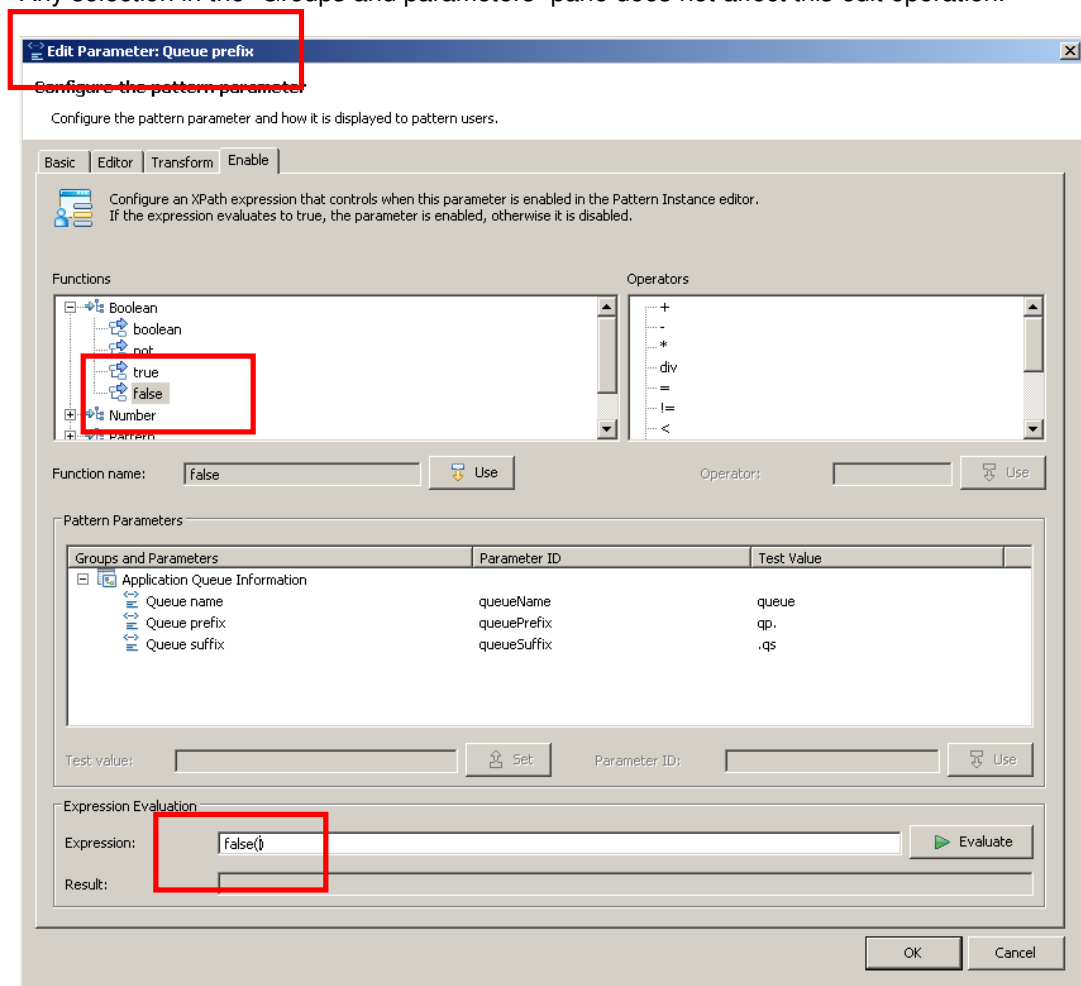
26. Click the Enable tab.

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

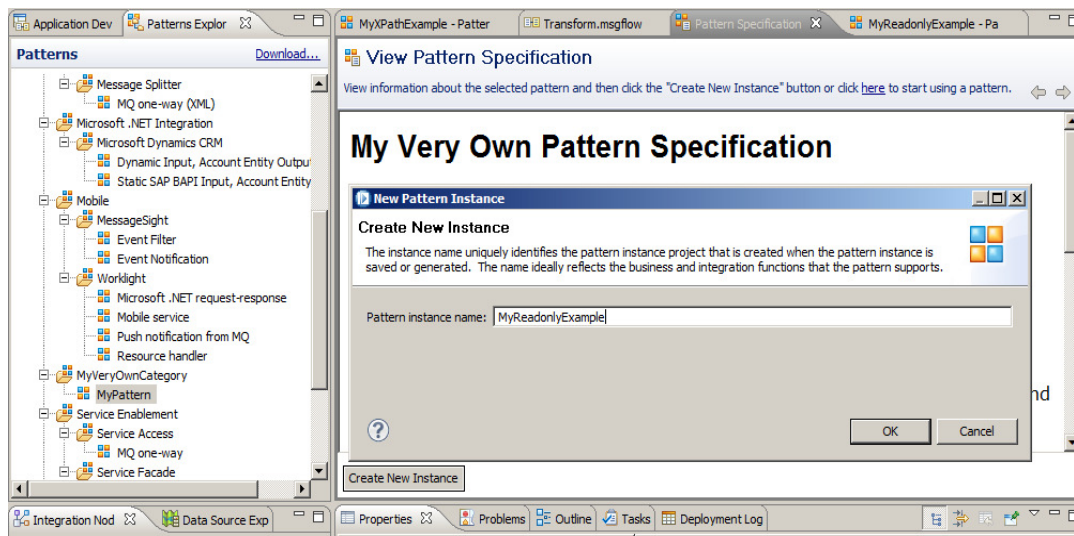
This sets the XPath expression to “false”, which tells the pattern to make the parameter un-editable.

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. Save the pattern.
29. Create the Patterns Plug-ins as before, then test the pattern again. Accept the default workspace.
30. In the new workbench, switch to the Patterns Explorer view.
31. Click MyPattern, and Create New Instance. Provide a name for the pattern instance, for example MyReadOnlyExample.



32. 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.

The screenshot shows a web-based configuration interface titled "Configure Pattern Parameters". At the top, there are two tabs: "Pattern specification" and "*MyReadOnlyExample - Pattern Configuration". Below the tabs, the main heading is "Configure Pattern Parameters". A message states: "Provide values for pattern parameters. Click the 'Generate' button or click [here](#) to generate a pattern instance." Below this, an information icon (i) says: "Pattern parameters are ready. Click the 'Generate' button to generate a pattern instance." The main content area is titled "Pattern Parameters" and contains a section for "Application Queue Information" which is expanded. Under this section, there is a label "Your own description" with a green checkmark. Three input fields are present: "Queue name *" with the value "queue", "Queue prefix *" with the value "qp.", and "Queue suffix *" with the value ".qs". The "Queue prefix" and "Queue suffix" fields are greyed out. At the bottom left of the main area is a "Generate" button. On the right side, there is a "Pattern Parameters Details" panel with a sub-section for "Application Queue Information". At the bottom of the dialog, there are two tabs: "Specification" and "Configuration".

Close the second workbench.

33. 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 name

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

Basic | Editor | Transform | Enable

Parameter Display
Display name: Queue 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.
 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 name

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>Describe the main part of the queue name</p>

Describe the main part of the queue name

This concludes the Pattern Authoring XPath lab.