



IBM Integration Bus

Pattern Authoring Lab 4

Adding and tailoring nodes using PHP and Java

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 create a pattern which enables a pattern user to dynamically change the message flow logic, based on the value of pattern parameters. The pattern that you create will include a new Compute node and associated ESQL which will be added to the generated message flow if the pattern selects the "database logging" option. The new node will take the entire contents of the message tree and insert it into a database row.

Two Pattern Authoring primary functions are used:

- 1. PHP scripts are used to create to the ESQL file for a new Compute node, and to create the related MQSC and DDL scripts to create MQ queues and database definitions.
- 2. A Java application, which creates the new Compute itself, and adds it to an existing message flow, depending on the value of the pattern parameters.

This hands-on lab is the fourth in a series of labs demonstrating the Pattern Authoring tools in IBM Integration Bus V9.0. Whilst this lab does not require any of the artefacts created in the first three labs, it is strongly recommended that you perform these earlier labs, before performing this lab. The reader is assumed to be familiar with the basic pattern authoring techniques explained in the earlier labs, and they are not covered here.

This lab uses a new message flow (exemplar) as its starting point; this is provided in the c:\student\PatternAuth directory, and is also packaged as a project interchange file for installation on the reader's own installation.

Specific Tasks

The lab comprises the following key steps:

- 2.1 Import the message flow (exemplar) and create the Pattern project.
- 2.2 Configure the pattern to add a new Compute node.
 - Create a PHP script to create the associated ESQL module
 - Create a Java application which creates the Compute node and adds it to the flow
- 2.3 Create a new pattern category
- 2.4 Test the pattern
- 2.5 Extend the pattern to add further PHP scripts which will create MQSC and DDL scripts.
- 2.6 Test the pattern again

Source files

This lab provides a number of source files. These are located in the folder c:\student\PatternAuth\resources.

2. Lab Guide

2.1 Create the pattern and add new pattern parameters

1. To avoid any naming conflicts, remove all projects from the earlier Pattern Authoring labs, including pattern projects, from your workspace (or create a new workspace).

Ensure you remove any associated pattern projects under "Independent Resources".

2. Import the PI file c:\student\PatternAuth\resources\PatternsAdvancedStartV8.zip. This will create a project named Transform_MFP.

Application Dev 🐹 🥰 Patterns Explor	ransform.msgflow 🛛
Application Dev 23 Patterns Explor Application Development Point Transform_MFP Point Transform_mgflow Point magi Point magi Point magi Point magi Point magi Point magi Point ma	ransform.msgflow ≥ Palette VeDSphere MQ MS MS MS MS MS MS MS MS MS MS
3,55 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	MQ Input CreateOutput MQ Output MQ Input CreateOutput MQ Output Iransformation Instruction Instabase Interference Interfer
	mail Log queue output CPIP IORBA Usiness Decisions ICS MS V
Grap	h User Defined Properties

Open the message flow, named Transform.msgflow.

3. Create a new Pattern Authoring project.



Pattern Authoring - adding nodes using PHP and Java Provided by IBM BetaWorks 4. Name the pattern PHPJavaPattern (this name will match code later in the lab).

Name the project PHPJavaPattern.

Click Next.

🕀 New Pattern	Authoring Project	
Create a New	Pattern Authoring Project	
This wizard creat	tes a new Pattern Authoring project.	
Pattern name:	PHPJavaPattern	
Project name:	PHPJavaPattern	
🔽 Lise default	t location	
Location: C:\u	workspaces\TSS\PHPlavaPattern	Browse,
<u>,</u>		
0	< Back Next > Fin	ish Cancel

5. Select the Transform_MFP referenced project, and click Finish.

🕀 New Pattern Authoring Pro	ject			
Create a New Pattern Au	thoring Proj	ect		
Specify dependencies on other p	projects.			
Referenced projects:				
Transform_MFP				
0	< Back	Next >	Finish	Cancel

6. In the new pattern project, select the "Pattern Configuration" tab.

attern Configuration gure your groups and pattern parameters and associate the pattern parameters i	wth their target properties.
pattern parameters and your own logic, you can extend your pattern with $\underline{J_{ava,\delta}}$	and PHP code.
ios and Parameters	
	💽 Add Group
	😂 Add <u>H</u> arameter
	Edt
	Delete
	Enumerated Types.
	Tabbs
	<u>2</u> 2
	Expand Al
	Collapse Al
and the code	

7. Create a new pattern group (click Add Group), and name it "Database Logging".

🖥 Add Group		x
Configure group Configure the pattern parameter grou Configure an XPath expression that co	up and how it is displayed to pattern users. ontrols when this group is enabled in the Pattern Instance editor.	
Basic Enable Group Display	Database Logging	
Description:	Pattern Parameters	
Generate help documentation	Select this option to create help information for this group in the pattern documentation.	
☑ Display parameters in a group b	ox Select this option to display this group, and any parameters it contains, with a surrounding box.	

Click OK.

Page 6

Pattern Authoring - adding nodes using PHP and Java Provided by IBM BetaWorks 8. Add a new pattern parameter (click Add Parameter). Set the following values:

Display name: Database Logging required Parameter ID: loggingRequired (case sensitive)

Add Parameter			
C onfigure pattern parameter Configure the pattern parameter and how	it is displayed to pattern users.		
Basic Editor Transform Enable			
Parameter Display Display name:	Database Logging required	Parameter ID:	loggingRequired
Parameter Options			
T Hide the parameter	Select this option to hide the parameter from the Use an XPath expression to set the value of the p	pattern user. arameter when a pattern instance is created.	
☑ Mandatory parameter	Select this option if the pattern user must enter a Mandatory parameters also display a field prompt	value for the parameter. to guide the pattern user.	
Field prompt:	Enter your parameter value		

 On the Editor tab, set the Parameter Editor to "Check Box", and the default value to true. Click OK.

🚔 Add Parameter		×
Configure pattern parameter Configure the pattern parameter and how it i	s displayed to pattern users.	
Basic Editor Transform Enable		
Parameter editor:	CheckBox (True/False)	nfigure Editor
Type selection:	Enumerated Types	Tabes
Default vaue:	true	

 Now add another pattern parameter, as follows: On the Basic tab: Display name: Logging database Parameter ID: loggingDatabase

On the Editor tab, set default value to SAMPLE

Click OK.

<->	Add Davamator			~
D	Difigure pattern parameter Configure the pattern parameter and how	t is displayed to pattern users.		<u>^</u>
ſ	Basic Editor Transform Enable Parameter Display Display name:	Logging database	Parameter ID:	loggingDatabase
	Parameter Options	Select this option to hide the parameter from the pa Use an XPath expression to set the value of the pa	ittern user. rameter when a pattern instance is created.	
	☑ Mandatory parameter	Select this option if the pattern user must enter a va Mandatory parameters also display a field prompt to	alue for the parameter. guide the pattern user.	
	Field prompt:	Enter your parameter value		

11. Similarly, add pattern parameters for database schema and table name:

Display name = Logging database schema Parameter ID = loggingSchema Default value = wmbadmin

Display name = Logging table Parameter ID = loggingTable Default value = LOG_TABLE

2.2 Configure the pattern to add a new Compute node

This section shows you how to add a new node with the pattern authoring tools.

We will add a Compute node to the flow (setting a couple of node properties), and create the associated ESQL file. This node will write the contents of the message tree to a single column in a database table. It will therefore need to have an associated ESQL file, and have the Database source node property set.

1. We will first create the ESQL module that the Compute node will reference. This is done with a PHP script.

On the Pattern Configuration tab, move down to the bottom of the editor, to the "Java and PHP code" section.

Click Add to add some PHP or Java code.

Transform.msgflow	🔡 *PHPJavaPattern.pattern 🔀	
Java and PHP code Using pattern parameters	and your own logic, you can extend your pattern with Java and PHP code. Use Java to access and up	ace the nodes, connections, and
properties in message flow	ws. Use PHP to generate text files, for example ESQL and MQSC scripts.	Add
		Edt
		Delete
		2 3
Source Files Pattern Configu	uration Categories Create Pattern	

2. Set the "Type of code" to PHP, and click "New Project".

dd code to your pat	tern
Select the code that is calle	ed when a pattern instance is generated.
ava or PHP	
When you create your patt To create a project for you	ern archive, your Java and PHP code projects are automatically packaged with your pattern plug-ins. r Java and PHP code, click "New Project"
ro creace a projeccitor you	
Turna of codes	
type of code;	
voiost pamo	- New Preject
rojecchane;	
Jua-in IDr	
ing in 10,)
ava	(
'he list in the "Class name"	field, displays the Java classes that you can use.
insure that the Java class	is exported from the code project.
'n evente e envir leven dens	that can be used in your pattern, click "New Java Class"
ro creace a new Java class	and can be asea in your pattern, allow new sava class .
io create a new Java tiass	
ilass name:	New Jaya Class
lass name:	New Java Class
lass name:	New Java Class
Class name: 21ass name: 21hP 21hoose the PHP file that ru	New Java Class
Class name: HP Choose the PHP file that ru You can choose to write the	New Java Class
Class name: HP Choose the PHP file that ru You can choose to write the To convert a file in your ref	New Java Class New Java Cla
Elass name: HP Choose the PHP file that ru You can choose to write the Fo convert a file in your ref	New Java Class New Java Cla
Eless name: HP Choose the PHP file that ru You can choose to write the Fo convert a file in your ref	New Java Class New Java Class New Java Class New Java Class Soutput from the PHP file into a file in a pattern instance project. erenced projects into a PHP file in your code project, click "Create From File". Conto From File
Elass name: HP Choose the PHP file that ru 'ou can choose to write the 'o convert a file in your ref	New Java Class New Java Class New Java Class New Java Class Create From File Create From File Create From File
Elass name: HP Choose the PHP file that ru You can choose to write the fo convert a file in your ref PHP file name:	New Java Class Create From File into a file in a pattern instance project. erenced projects into a PHP file in your code project, click "Create From File". Create From File Create From File
Class name: HP Choose the PHP file that ru You can choose to write the fo convert a file in your ref PHP file name: Write the output from t	New Java Class Create From File into a file in a pattern instance project. Create From File he PHP file into an output file:
Class name: HP Choose the PHP file that ru You can choose to write the fo convert a file in your ref PHP file name: Write the output from t Dutput project name:	New Java Class Create From File into a file in a pattern instance project. Create From File Create From File PHP file into an output file: Transform_MFP
Class name: HP Choose the PHP file that ru You can choose to write the fo convert a file in your ref PHP file name: Write the output from t Dutput project name:	New Java Class Create in stance is created. Create From File". Create From File Create From File Transform_MFP
Class name: Class name: Choose the PHP file that ru YHP file name: YHP file name: Write the output from t Dutput project name: Dutput file name:	New Java Class Create from the PHP file into a file in a pattern instance project. erenced projects into a PHP file in your code project, click "Create From File". Create From File Create From File Transform_MEP
Class name: Class name: Choose the PHP file that ru fou can choose to write the fo convert a file in your ref PHP file name: Write the output from t Dutput project name: Dutput file name:	New Java Class Create into a file in a pattern instance project. erenced projects into a PHP file in your code project, click "Create From File". Create From File Create From File Transform_MEP
Class name: Class name: Choose the PHP file that ru YHP Choose the PHP file that ru YHP file name: Write the output from t Dutput project name: Dutput file name:	New Jave Class Create From File into a file in a pattern instance project. Create From File Create From File Transform_MEP Create Case

3. Accept the defaults and click Finish.

🕀 New Pattern Autho	pring Java and PHP Project	<u>_ 0 ×</u>
New pattern autho	oring Java and PHP project	1 I
Configure the pattern a	authoring Java and PHP project.	
-Plug-in		
This wizard creates a	pattern authoring project.	
The project is a plug-ir it is loaded by the Wel	n that you package and distribute with your pattern plug-ins. The Plug-in ID uniquely identifie bSphere Message Broker Toolkit.	s the plug-in when
The Plug-in ID is also u	used to name the project in your workspace.	
Plug-in ID:	com.your.company.domain.PHPJavaPattern.code	
Java		0
Pattern authoring code Use Java to manipulate Java can also implemen	e projects always support Java. e message flows in pattern instance projects. nt other custom logic required by your pattern.	Ŭ
🔽 Add an example pa	attern authoring Java class to the project	
Package name:	com.your.company.domain.PHPJavaPattern.code	
Class name:	МуЈаva	
Use PHP to generate to If you enable this optic	ext files, such as MQSC and ESQL scripts, in pattern instances projects. on, a templates directory and PHP examples are added to your project.	
🔽 Add PHP support to	o the project	
0	Finish	Cancel

4. Note that the PHP file name has been set to templates/main.php. We will leave it for this lab, but you can change this as required.

Click OK.

ld code to your p	attern		
select the code that is (called when a pattern instance is generated.		
ava or PHP			
When you create your p To create a project for y	pattern archive, your Java and PHP code projects are automatically packaged v your Java and PHP code, click "New Project".	with your pattern p	lug-ins.
Type of code:	РНР		
Project name:	com.your.company.domain.PHPJavaPattern.code	_	New Project
Plug-in ID:	com.your.company.domain.PHPJavaPattern.code		
lava			
ava he list in the "Class nan nsure that the Java cla o create a new Java cl lass name:	me" field, displays the Java classes that you can use. ass is exported from the code project. ass that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava		New Java Class
lava The list in the "Class nan Ensure that the Java de Fo create a new Java d Class name: PHP Choose the PHP file that You can choose to write Fo convert a file in your	me" field, displays the Java classes that you can use. ass is exported from the code project. lass that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava com.your.company.domain.PHPJavaPattern.code.MyJava t runs when a pattern instance is created. t the output from the PHP file into a file in a pattern instance project. referenced projects into a PHP file in your code project, click "Create From File		New Java Class
Java The list in the "Class nan Ensure that the Java de To create a new Java d Class name: PHP Choose the PHP file that You can choose to write To convert a file in your PHP file name:	me" field, displays the Java classes that you can use. ass is exported from the code project. lass that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava t runs when a pattern instance is created. t the output from the PHP file into a file in a pattern instance project. referenced projects into a PHP file in your code project, click "Create From File templates/main.php		New Java Class
Java The list in the "Class nan Ensure that the Java cla To create a new Java cl Class name: PHP Choose the PHP file that You can choose to write To convert a file in your PHP file name: Write the output fro	ne" field, displays the Java classes that you can use. ass is exported from the code project. lass that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava t runs when a pattern instance is created. t the output from the PHP file into a file in a pattern instance project. referenced projects into a PHP file in your code project, click "Create From File templates/main.php m the PHP file into an output file:		New Java Class., Create From File.,
Java The list in the "Class nam Ensure that the Java de To create a new Java de Class name: PHP Choose the PHP file that You can choose to write To convert a file in your PHP file name: Write the output fro Output project name;	me" field, displays the Java classes that you can use. ass is exported from the code project. lass that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava t runs when a pattern instance is created. t the output from the PHP file into a file in a pattern instance project. referenced projects into a PHP file in your code project, click "Create From File templates/main.php m the PHP file into an output file: Transform_MFP		New Java Class., Create From File.
Java The list in the "Class nan Ensure that the Java de To create a new Java de Class name: Choose the PHP file that You can choose to write To convert a file in your PHP file name: Write the output fro Output project name; Output file name;	ne" field, displays the Java classes that you can use. ass is exported from the code project. lass that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava t runs when a pattern instance is created. t the output from the PHP file into a file in a pattern instance project. referenced projects into a PHP file in your code project, click "Create From File templates/main.php m the PHP file into an output file: Transform_MFP		New Java Class.,

5. In the project navigator, under Independent Resources, expand the src and templates directories, as shown.



6. Open the main.php file.

Delete the entire contents of this file.

Replace the contents with the entire contents of the file c:\student\PatternAuth\resources\main.php.txt.

📕 n	nain.php.txt - Notepad
File	Edit Format View Help
p</th <th>hp</th>	hp
	phpinfo();
	<pre>\$today = getdate(); print_r(\$today);</pre>
	<pre>var_dump(\$_MB);</pre>
?>	<pre>if (\$_MB['PP']['loggingRequired'] == 'true') { mb_pattern_run_template("Transform_MFP", "mqsi/logging.esql.php", "mqsi/LogToDatabase.esql"); }</pre>

Things to observe:

- a) The "if" statement checks the value of the loggingRequired pattern parameter.
- b) The mb_pattern_run_template statement references the "Transform_MFP" project.
- c) It runs the mqsi/Logging.esql.php script (we'll create this shortly)
- d) The output is written to the file mqsi/LogToDatabase.esql.

Save and close main.php.

Rename the file example.<u>esql</u>.php to logging.esql.php (under templates\mqsi).
 Note: make sure you re-name the correct file

	×
ОК	Cancel
	ОК

8. Open the file logging.esql.php, and replace the entire contents of this file with the contents of c:\student\PatternAuth\resources\logging.esql.php.txt.

Note that the Broker Schema, mqsi, has been hard-coded into the PHP script. This can be replaced with a variable value if required, of omitted completely if you are going to use the default schema.

```
🕫 Transform.msgflow 🗧 *PHPJavaPattern.pattern 📄 logging.esql.php 🔀
 BROKER SCHEMA mosi
  DECLARE ns NAMESPACE 'http://<?php echo %_MB['PP']['loggingDatabase']; ?>/<?php echo %_MB['PP']['loggingSche
  CREATE COMPUTE MODULE LogToDatabase
      CREATE FUNCTION Main() RETURNS BOOLEAN
      BEGIN
      DECLARE msgBitStream BLOB InputRoot.BLOB.BLOB;
      INSERT INTO Database.<?php echo $_MB['PP']['loggingSchema']; ?>.<?php echo $_MB['PP']['loggingTable'];</pre>
          RETURN TRUE;
      END;
      CREATE PROCEDURE CopyMessageHeaders() BEGIN
          DECLARE I INTEGER 1;
          DECLARE J INTEGER;
          SET J = CARDINALITY(InputRoot.*[]);
          WHILE I < J DO
             SET OutputRoot.*[I] = InputRoot.*[I];
              SET I = I + 1;
          END WHILE;
      END:
      CREATE PROCEDURE CopyEntireMessage() BEGIN
         SET OutputRoot = InputRoot;
      END;
 END MODULE;
```

A number of the static values have been replaced with variables based on PHP strings: a) <?php echo \$_MB[MB['PP']['loggingDatabase']; ?> b) <?php echo \$_MB[MB['PP']['loggingSchema']; ?> c) <?php echo \$_MB[MB['PP']['loggingTable']; ?>

Save and close the PHP file.

9. Now we're going to add a new Compute node into the flow. This node will use the ESQL file we just created. To add a new node, we will use a small amount of java code.

In the Pattern Configuration, click Add to add a new Java item.



Source Files Pattern Configuration Categories Create Pattern

10. We will use the same project that we created earlier, so just click OK.

Select the code that is ca	alled when a pattern instance is generated.		
Java or PHP			
When you create your pa To create a project for yo	ittern archive, your Java and PHP code projects are automatically packaged with our Java and PHP code, click "New Project".	your pattern p	olug-ins.
Type of code:	Java		
Project name:	com.your.company.domain.PHPJavaPattern.code	•	New Project
Plug-in ID:	com.your.company.domain.PHPJavaPattern.code		
Java The list in the "Class name Ensure that the Java clas To create a new Java clas	a" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class".		
Java The list in the "Class name Ensure that the Java class To create a new Java class Class name:	a" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava	T	New Java Class
Java The list in the "Class name Ensure that the Java clas To create a new Java clas Class name: PHP	a" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava	×	New Java Class
Java The list in the "Class name Ensure that the Java class To create a new Java class Class name: PHP Choose the PHP file that r You can choose to write the To convert a file in your re	a" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava runs when a pattern instance is created. he output from the PHP file into a file in a pattern instance project. eferenced projects into a PHP file in your code project, click "Create From File".	T	New Java Class
Java The list in the "Class name Ensure that the Java class To create a new Java class Class name: PHP Choose the PHP file that r You can choose to write the To convert a file in your re PHP file name:	e" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava runs when a pattern instance is created. he output from the PHP file into a file in a pattern instance project. eferenced projects into a PHP file in your code project, click "Create From File". templates/main.php	Y	New Java Class
Java The list in the "Class name Ensure that the Java class To create a new Java class Class name: PHP Choose the PHP file that r You can choose to write ti To convert a file in your re PHP file name: Write the output from	a" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava runs when a pattern instance is created. he output from the PHP file into a file in a pattern instance project. eferenced projects into a PHP file in your code project, click "Create From File". templates/main.php the PHP file into an output file:	•	New Java Class Create From File
Java	a" field, displays the Java classes that you can use. s is exported from the code project. ss that can be used in your pattern, click "New Java Class". com.your.company.domain.PHPJavaPattern.code.MyJava runs when a pattern instance is created. he output from the PHP file into a file in a pattern instance project. eferenced projects into a PHP file in your code project, click "Create From File". templates/main.php the PHP file into an output file: Transform_MFP	Y	New Java Class Create From File

11. In the code project, expand all items under the src directory, and open MyJava.java.

Replace the entire contents of this file with the entire contents of c:\student\PatternAuth\resources\loggingNode.java.txt.

Take a few minutes to look at the key code statements, which is well commented.

This java code uses the new Message Flow API, which allows you to manipulate a message flow, add/remove nodes and connectors, and set node properties.

Save and close Myjava.java.

package com.your.company.domain.PHPJavaPattern.code;
import com.ibm.broker.config.appdev.ESQLModule; import com.ibm.broker.config.appdev.MessageFlow; import com.ibm.broker.config.appdev.nodes.ComputeNode; import com.ibm.broker.config.appdev.patterns.GeneratePatternInstanceTransform;
import com.ibm.broker.config.appdev.patterns.PatterninstanceManager;
public class MyJava implements GeneratePatternInstanceTransform {
@Override public void onGeneratePatternInstance(PatternInstanceManager patternInstanceManager) { // Get the name of the message flow MessageFlow mf1 = patternInstanceManager.getMessageFlow("Transform_MFP", "mqsi/Transform.msgflow");
// Get the value of the loggingrequired pattern parameter String loggingNeeded = patternInstanceManager.getParameterValue("loggingRequired");
<pre>// Has database logging been selected by the pattern user? if (loggingNeeded.equals("true") == true) {</pre>
// Specify the schema and file name for the associated ESQL file (created in our PHP script)
ESQLModule myEsqlModule = new ESQLModule(); myEsqlModule.setBrokerSchema("mqsi"); myEsqlModule.setEsqlMain("LogToDatabase");
<pre>// Create a new Compute node called LogToDatabase, and specify its position on the flow editor ComputeNode loggingNode = new ComputeNode(); loggingNode.setNodeName("LogToDatabase"); loggingNode.setLocation(400,50);</pre>
<pre>// Obtain the value of the pattern parameter for the logging database, and use this to // set the database source property on the new Logging node. String loggingDB = patternInstanceManager.getParameterValue("loggingDatabase"); loggingNode.setDataSource(loggingDB);</pre>
<pre>// Set the ESQL file name property on the new Logging node</pre>
// Add the new node to the flow mf1.addNode(loggingNode);
<pre>// Get a handle on the existing node in the flow, called "CreateOutput", and // create a connector from this node to our new Logging node. ComputeNode createoutputNode = (ComputeNode) mf1.getNodeByName("CreateOutput"); mf1.connect(createoutputNode.OUTPUT_TERMINAL_OUT, loggingNode.INPUT_TERMINAL_IN);</pre>
}

Pattern Authoring - adding nodes using PHP and Java Provided by IBM BetaWorks

2.3 Create a new Pattern Category

1. Now switch to the Categories tab. Create a new category (Add Category), and name it "PHP and Java Examples".



2. Drag the new pattern onto the new category.



2.4 Test the Pattern

1. Switch to the "Create Pattern" tab, and click Test Pattern.

🕫 Transform.msgflow 🔒	*PHPJavaPatt	ern.pattern 🛛		
进 Create Pattern				
Test your pattern by configu	ring your patterr	n plug-in informati	ion, click "Test Pattern".	
🔗 Create Pattern	Plug-ins		🜔 Test Pattern	🕸 Debug Pattern
Plug-in Information				
🔡 Configure the unique	identifier for you	r pattern plug-in.		
Pattern name:	PHPJavaPatt	ern		
Plug-in ID:	com.your.co	mpany.domain.Pł	HPJavaPattern	
Version:	1.0.0.0			
Provider:	Your Compar	ny Name		
Description:	Plug-in creat	ed by the Patterr	Authoring editor	
Translation Options				
If you enable this option, t If you are creating a single Create translation plug	he Pattern Auth anguage patter nins (*.nl1 and *	oring editor creat rn, do not select (.doc.nl1)	es two additional NLS plug-ins. ` :his check box.	These plug-ins are set up so that you can drop in tr
Pattern Distribution				
After you have tested you	ır pattern plug-in	s, package them i	into a pattern archive by clicking	"Create Pattern Archive" and then distribute the
To install the pattern, the	pattern user mus	t launch the patt	ern archive file in a web browse	r, launch the pattern archive file in a file explorer,
For more information abou	t packaging and	distributing user-(defined patterns, click <u>here</u> .	
Source Files Dattern Configura	tion Categories	Create Pattern		

2. When the new workbench starts, select the Patterns Explorer, and click the PHPJavaPattern entry.



3. Click Create New Instance, provide a pattern instance name, and click OK.

4. Expand the "Database Logging" group, and observe the parameters and default values that you have just created.

You can change these values if you want (but leave the Database Logging Required box checked).

Click Generate.

Configure Pattern Parameters Provide values for pattern parameters. Click the "Generate" button or click Pattern workspace resources are generated successfully!	k <u>here</u> to generate a patter
Pattern Parameters	E 🗆 🖻
▼ Database Logging Pattern Parameters Database Logging required * Logging database * Logging database * Logging database schema * Wmbadmin Logging table *	
Generate Specification Configuration	

5. A new message flow will be generated (this example was generated with a pattern instance name of Test1).

Observe that a new node, LogToDatabase, has been added to the flow. Click this node and see that the corresponding ESQL module has been set to mqsi.LogToDatabase.

Observe that a new ESQL file, LogToDatabase.esql, has been created in the mqsi schema, in the project navigator.

Observe the new node Data source name = SAMPLE.

🗟 Application Dev 🛛 🐯 Patterns Explor 🗧 🗖	Pattern Specification	B logging_test - Pattern Configuration	Transform.msgflow	- 8
	👌 😳 Palette			
Application Development New.	. 🕟 🗸 🖉	•		
Pattern Instances	强 Favorites			
E Iogging_test_Transform_MFP	🔐 WebSphere MQ			
Flows	Gms JMS		(LogToDatabase)	
Transform monflow	💭 НТТР			
ESOLS	🧟 Web Services			
🖻 🙀 masi	😡 SCA			
E Sel LogToDatabase.esq	WebSphere Adapters	MQ Input	CreateOutput MQ Output	
±™ <u>es</u> i transform.esgi	GR Routing		\sim	
	💭 .NET			
	Transformation			
	Construction			
	C) Database			
	🚱 File		** @ F	
	Smp Email		Log queue output	
	CPIP			
	CORBA	-1		
	Graph User Defined Propert	ties		
	Properties 🛛 🖹 P	roblems 🔡 Outline 🧟 Tasks 🛄 Deployr	nent Log	
	🦑 Compute Node Pro	operties - LogToDatabase	· · · · · · · · · · · · · · · · · · ·	
	Description			
	Basic Data source	sample		
	Validation Connect b	efore flow starts		
	Monitoring	n Automatic		
	ESCI mod		hana	Browse
	ESQLINOU			
	Compute n	node Hiterrage		
	Treat warr	nings as errors		
127 Integration Nod 🕅 🔪 👪 Data Source Evo 📄 🦷 🖡	Ihrow exc	epuon on database error 🛛 🗹		1

6. Open the ESQL file (double-click either the new Compute node, or the ESQL file in the project navigator).

Observe that several variables have been set to the values specified by the pattern instance.

```
🖽 *Transform.msgflow
                                        📴 LogToDatabase.esql 🛛
                                                              »<sub>1</sub>
🔡 PHPJavaTest1 - Patte
 BROKER SCHEMA mosi
 DECLARE ns NAMESPACE 'http://SAMPLE/wmbadmin';
 CREATE COMPUTE MODULE LogToDatabase
     CREATE FUNCTION Main() RETURNS BOOLEAN
     BEGIN
     DECLARE msgBitStream BLOB InputRoot.BLOB.BLOB;
      INSERT INTO Database.wmbadmin.LOG TABLE VALUES (msgBitStream);
۵
          RETURN TRUE;
      END :
      CREATE PROCEDURE CopyMessageHeaders() BEGIN
          DECLARE I INTEGER 1;
          DECLARE J INTEGER;
          SET J = CARDINALITY(InputRoot.*[]);
          WHILE I < J DO
              SET OutputRoot.*[I] = InputRoot.*[I];
              SET I = I + 1;
          END WHILE:
      END :
      CREATE PROCEDURE CopyEntireMessage() BEGIN
          SET OutputRoot = InputRoot;
      END .
 END MODULE;
 ◀
```

Close the ESQL file, and the message flow.

7. Now we will create a new message flow, where we do not require database logging.

In the pattern instance, uncheck the "Database Logging Required" tickbox.

Note – we could have built the pattern to disable (grey out) the database fields if this option is deselected. We will leave that as an exercise for the reader – see the earlier patterns labs for details.

Click Generate.

Pattern Authoring - adding nodes using PHP and Java Provided by IBM BetaWorks

8. Open the new message flow.

Observe that the flow does not contain a database logging node.

Observe that the ESQL section only contains ESQL for the existing Compute node.



2.5 Using PHP to generate MQSC and DDL scripts

Now we'll go and add a couple of other types of scripts to the pattern. In this section, we'll create an MQSC script.

Close the test workspace, and return to the Pattern Editor.

1. First, we need to add some MQ queue names as pattern parameters.

You should know how to do this by now, so no screen captures.

In the Message Flow editor for Transform.msgflow, right-click each of the three MQ nodes in turn, and in each case select the queue name as a pattern parameter (don't select the queue manager name on the MQ output nodes by accident).

Save the flow.

In the PHPJavaPattern, select Pattern Configuration. You will see the new pattern parameters added to the list.

Add a new group, and give it a title "Queue Names". Drag each of the MQ queue name parameters into this group (drag the items named "Queue Name"). Note that the pattern editor has automatically created new individual groups for each of the MQ nodes (named MQ_Input, etc). You can now safely discard these (highlight each, click Delete).

For each of the queue name parameters, click Edit, and set the following values: $\ensuremath{\mathsf{MQ}}\xspace_{\ensuremath{\mathsf{Input}}\xspace}$ node

Display name = Input queue name Parameter ID = inputQueueName (case sensitive) Default value (on the Editor tab) = INPUT.QUEUE

MQ_Output node

Display name = Output queue name Parameter ID = outputQueueName Default value = OUTPUT.QUEUE

MQ_Log_output Display name = Log queue name Parameter ID = logQueueName

Default value = LOG.QUEUE

2. At this point, the pattern configuration should look like this:

Pattern Configuration Infigure your groups and pattern parameters and associate the pattern parameters with their target properties. Ing pattern parameters and your own logic, you can extend your pattern with <u>Java and PHP code</u> .	
Coups and Parameters Image: Detabase Logging Tequired (loggingRequired) Image: Detabase Logging Database (loggingDatabase) Image: Dogging database scheme (loggingSchema) Image: Dogging database scheme (logQueueName) Image: Image: Dog queue name (logQueueName) Image: Dogging databae Imag	Image: Add Group Image: Add Barameter Edit Delete Enumerated Types Tables Image: Add Barameter Expand All Cgllapse All

Save the pattern configuration.

3. Still in Pattern Configuration, move the window down to the "Java and PHP" section.

Click Add to add a new PHP script.

lava and PHP code	
Jsing pattern parameters and your own logic, you can extend your pattern with Java and PHP code. Use Java to ac properties in message flows. Use PHP to generate text files, for example ESQL and MQSC scripts.	cess and update the nodes, connections, and
Runs PHP file: templates/main.php (com.your.company.domain.PHPJavaPattern.code)	Add
🕞 Invokes Java class: com.your.company.domain.PHPJavaPattern.code.MyJava	
	Edit
	Delete
	2
rce Files Pattern Configuration Categories Create Pattern	

4. Set "Type of code" = PHP.

Set "PHP file name" = templates/scripts/example.mqsc.php (one of the drop-down options).

Select "Write output from PHP to an output file".

Set "Output project name" = Transform_MFP.

Click OK.

			-
Add code to your pa Select the code that is ca	ttern IIed when a pattern instance is generated.		
lava or PHD			
When you create your pa To create a project for yo	ttern archive, your Java and PHP code projects are automatically packaged with yo ur Java and PHP code, click "New Project".	ur pattern p	olug-ins.
Type of code:	РНР		7
Project name:	com.your.company.domain.PHPJavaPattern.code	•	New Project
Plug-in ID:	com.your.company.domain.PHPJavaPattern.code		
The list in the "Class name Ensure that the Java class To create a new Java class	" field, displays the Java classes that you can use. is exported from the code project. s that can be used in your pattern, click "New Java Class".		New Jaco Class
PHP Choose the PHP file that r You can choose to write t To convert a file in your re	uns when a pattern instance is created. ne output from the PHP file into a file in a pattern instance project. ferenced projects into a PHP file in your code project, click "Create From File".		ŝ.
PHP file name:	templates/scripts/example.mqsc.php	•	Create From File
🔽 Write the output from	the PHP file into an output file:		
Output project name:	Transform_MFP		•
Output file name:	scripts/example.mqsc		

5. Open the file templates/scripts/example.mqsc.php.

Replace the entire contents of the file with the contents of c:\student\PatternAuth\resources\Create_mqsc.php.txt.

Note that the queue names in this script have been set to the values of the queue name pattern parameters. (It's important that you typed the parameter IDs correctly when you added the queue parameters earlier).



Save and close.

 Now we will add a PHP script to create the database DDL statements to create the database logging table. We only want to run this script if "Database Logging" is selected by the pattern user.

Open the file main.php.

Inside the "if" clause, add a second line, which will invoke the CreateLT.ddl.php script file (we'll create this in the next step).

```
mb_pattern_run_template("Transform_MFP", "mqsi/CreateLT.ddl.php",
"mqsi/CreateLT.ddl");
```

(To avoid mistakes, you can copy from the file c:\student\PatternAuth\resources\main.php.line2.txt).

7. Under the templates/mqsi directory, import the file CreateLT.ddl.php. (Right-click, Import, General, File System, C:\student\PatternAuth\resources\CreateLT.ddl.php.

Open the newly imported file and observe that several fields have been set to PHP variables, matching the pattern parameters you created earlier.

8. Now we're done.

Save the pattern, then click Create Pattern, and Test Pattern to test the updated pattern in the usual way.

9. Create a new pattern instance. This example is named Test4.

You can override the default values if you want.

Click Generate.

Pattern Specification	est4 - Pattern Configuration 🕱		
🖏 Configure Pattern	Parameters		
Provide values for pattern paramete	ers. Click the "Generate" button or click <u>here</u> to generate a pa	ttern ins	tance.
i Pattern parameters are ready. C	lick the "Generate" button to generate a pattern instance.		
Pattern Parameters		Ŧ.	-
▼ Database Logging		1	
Pattern Parameters			
Database Logging required *			
Logging database *	SAMPLE		
Logging database schema *	wmbadmin		
Logging table *	LOG_TABLE		
▼ Queue Names		1	
Pattern Parameters			
Output queue name *	OUTPUT.QUEUE		
Input queue name *	INPUT.QUEUE		
Log queue name *	LOG.QUEUE		
(-
Generate			
pecification Configuration			

10. Open the scripts/CreateLT.ddl file.

Observe the database, schema and table names have been set to those specified in the pattern instance.



11. Open the scripts/example.mqsc file.

Observe the queue names (and descriptions) have been set to those specified in the pattern instance.



This concludes the Pattern Authoring PHP and Java lab.