



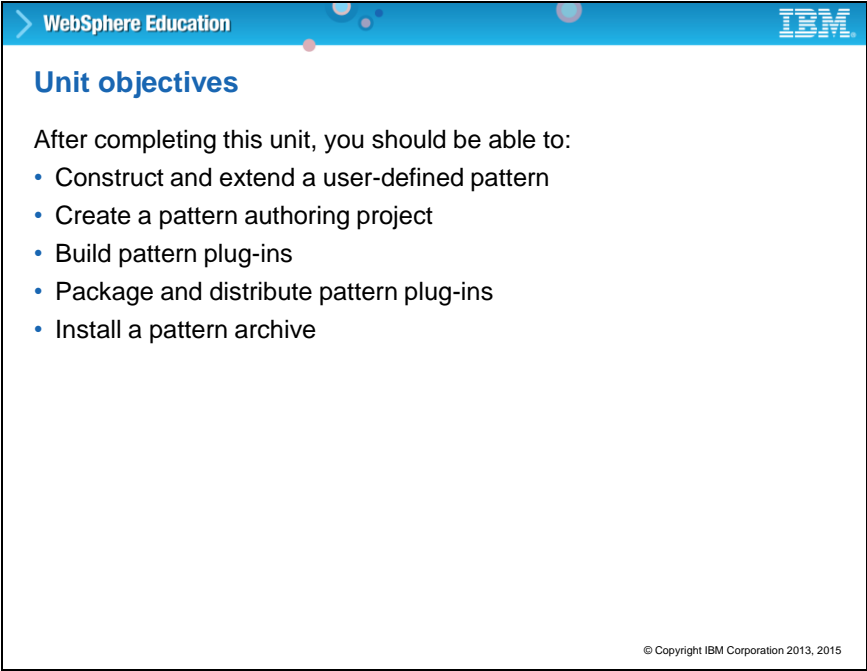
Slide 1


WebSphere Education 

Creating patterns for reusability



© Copyright IBM Corporation 2013, 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

A presentation slide with a blue header bar containing the text 'WebSphere Education' and the IBM logo. The main content area is white and contains the title 'Unit objectives' in blue, followed by a paragraph and a bulleted list. A small copyright notice is at the bottom right.

> WebSphere Education 

Unit objectives

After completing this unit, you should be able to:

- Construct and extend a user-defined pattern
- Create a pattern authoring project
- Build pattern plug-ins
- Package and distribute pattern plug-ins
- Install a pattern archive

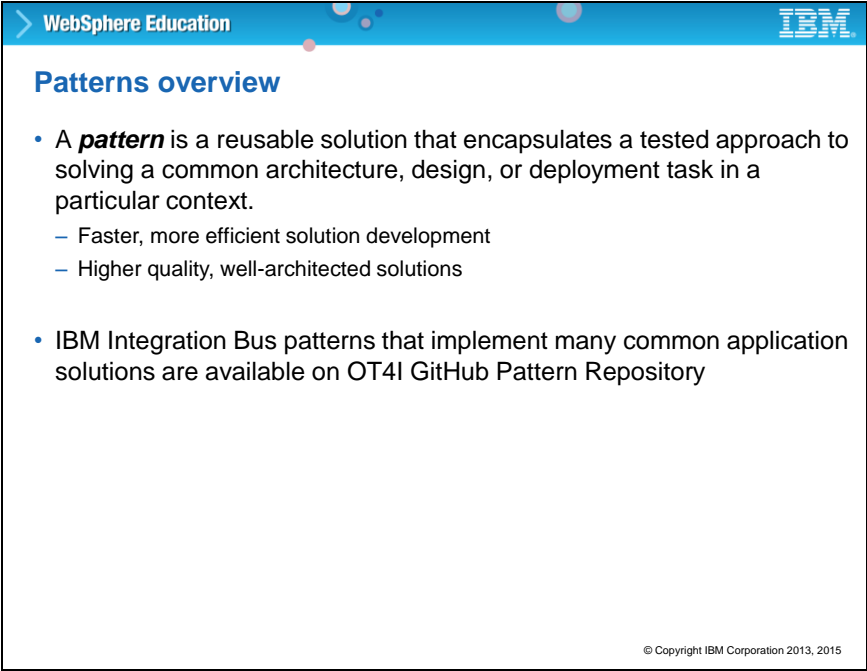
© Copyright IBM Corporation 2013, 2015

Unit objectives


This unit describes how to create and test user-defined message flow patterns in the IBM Integration Toolkit.

After completing this unit, you should be able to:

- Construct and extend a user-defined pattern
- Create a pattern authoring project
- Build pattern plug-ins
- Package and distribute pattern plug-ins
- Install a pattern archive



The slide is titled "Patterns overview" and is part of a "WebSphere Education" presentation, as indicated by the header. It features the IBM logo in the top right corner. The main content is a bulleted list defining patterns and their availability. The footer contains a copyright notice for IBM Corporation from 2013 to 2015.

> WebSphere Education 

Patterns overview

- A **pattern** is a reusable solution that encapsulates a tested approach to solving a common architecture, design, or deployment task in a particular context.
 - Faster, more efficient solution development
 - Higher quality, well-architected solutions
- IBM Integration Bus patterns that implement many common application solutions are available on OT4I GitHub Pattern Repository

© Copyright IBM Corporation 2013, 2015

Patterns overview

A pattern captures a tested solution to a common application problem or specific business problem.

When you start your Integration Bus development by using a pattern:

- Your applications are based on well-designed solutions that are analyzed and optimized.
- You can decrease your development time because you are not creating the project “from scratch”.
- You can learn preferred techniques for developing application solutions.

Patterns-based development takes advantage of common practices by standardizing message flows. It eases some of the anxiety that new users can experience when presented with so many node choices, and it gives a quick start to message flow design. A pattern reduces common problems by providing templates that solve common problems and promote good practices.

A catalog of Integration Bus patterns is available on an external GitHub repository. You select and download the pattern that meets your development needs to your Integration Toolkit.

WebSphere Education

IBM

Using patterns from the GitHub repository

1. Download pattern from OT4I GitHub Pattern Repository
2. Review the **Pattern Specification** to:
 - Learn what the pattern does
 - See constraints or limitations on using the pattern
 - Review any prerequisites tasks, constraints, and limitations
 - Understand the tasks that must be completed after implementing the pattern
3. Click **Create New Instance** in the Patterns Explorer
4. Configure the pattern by setting *pattern parameters*
5. Generate a *pattern instance project*

Patterns

- Patterns
 - File Processing
 - Record Distribution
 - HTTP one-way
- Pattern Repositories
 - OT4I GitHub Pattern Repository
 - Samples
 - Experimental OT4I GitHub Pattern Repository
 - File Processing
 - Record Distribution
 - HTTP one-way [Installed]
 - Mobile

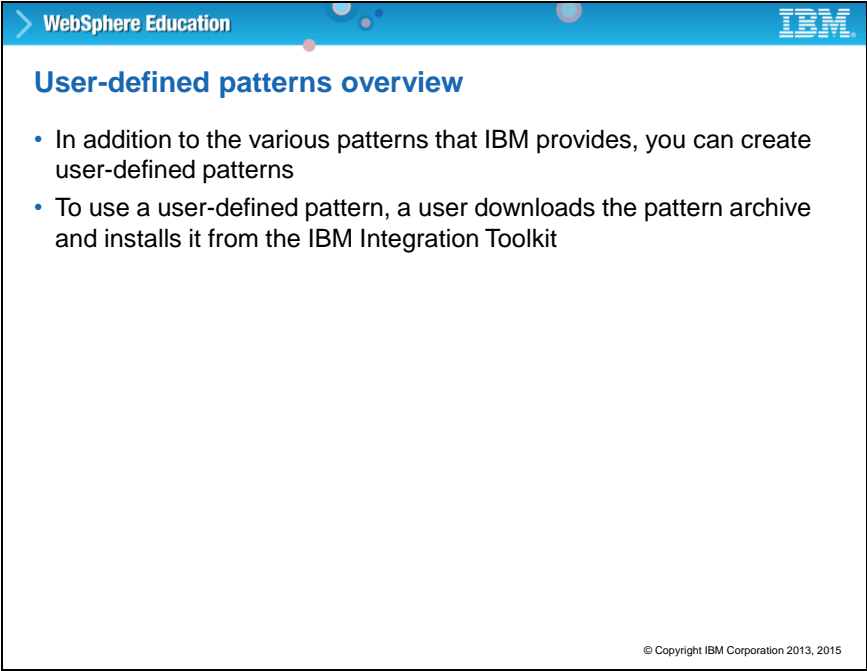
© Copyright IBM Corporation 2013, 2015

Using patterns from the GitHub repository


The Patterns Explorer lists the available patterns in the GitHub repository and those patterns that were downloaded to the Integration Toolkit.

When you elect to implement a pattern, you follow a standard defined sequence of steps. Finally, these steps include creating an instance of the template pattern, then configuring that instance, and generating the pattern instance. This slide lists the steps for getting patterns from the GitHub Pattern Repository and that uses the pattern to create a pattern instance.

These steps are used for built-in patterns and user-defined patterns after they are installed in the Patterns Explorer.



The slide features a blue header bar with the text 'WebSphere Education' on the left and the IBM logo on the right. The main content area is white with a blue title 'User-defined patterns overview'. Below the title are two bullet points. The footer contains a small copyright notice.

> WebSphere Education 

User-defined patterns overview

- In addition to the various patterns that IBM provides, you can create user-defined patterns
- To use a user-defined pattern, a user downloads the pattern archive and installs it from the IBM Integration Toolkit

© Copyright IBM Corporation 2013, 2015

User-define patterns overview

Like many components of Integration Bus, the concept of patterns is extensible. In addition to taking advantage of a number of built-in patterns, you can also create user-defined patterns.

After you create a user-defined pattern, you can package and distribute it so that another developer can download the user-defined pattern to their Toolkit installation.

User-defined patterns can help you to extend good practices within your organization by providing predefined templates from which to start Integration Bus application development.

WebSphere Education

IBM

Process for creating a user-defined pattern

| | |
|--|--|
| Integration developer <ul style="list-style-type: none">• Develops exemplar message flow• Tests pattern | Pattern user <ul style="list-style-type: none">• Installs pattern• Browses Pattern Explorer• Enters pattern parameters• Generates pattern instance• Customizes pattern instance• Deploys pattern instance• Tests pattern instance |
| Pattern author <ul style="list-style-type: none">• Defines target properties• Selects files and resources• Defines pattern user instance• Build pattern plug-ins• Tests pattern• Distributes pattern | |

© Copyright IBM Corporation 2013, 2015

Process for creating a user-defined pattern



The Integration Bus developer, a pattern author, and a pattern user complete the three stages of creating a user-defined pattern.

The Integration Bus developer develops the exemplar message flow. The exemplar is fundamental to the pattern authoring process as it defines the base message flow and is the starting point for a pattern.

The pattern author creates a pattern plug-in from the exemplar. Some of the pattern authoring tasks focus on the resources in Integration Bus. For example, one step in authoring a pattern is defining the target properties for the pattern and configuring the user interface that is presented to the pattern user.

The pattern author ensures that the pattern users can customize the pattern as needed. If the pattern user regenerates an instance of the pattern, the pattern author also ensures that these customizations are not overwritten. The pattern author then shares the user-defined pattern with the pattern user.

The pattern user is the Integration Bus developer that receives a user-defined pattern, generates a pattern instance, and uses it in accordance with the requirements of the organization.



Creating a user-defined pattern

- Integration developer creates an exemplar message flow that reflects good practice for the organization
 - Exemplar pattern contains components that are needed to define the base content for the pattern
 - Use the Pattern Authoring editor in the Integration Toolkit to create and edit a pattern authoring project
- Pattern author develops pattern plug-ins and generated documentation for the pattern plug-ins
- Integration developer and pattern author test the pattern plug-ins and the generated pattern instance
- Pattern author packages the pattern plug-ins into a pattern archive
- Pattern author distributes the pattern plug-ins to individual users or a patterns community site

© Copyright IBM Corporation 2013, 2015

Creating a user-defined pattern

The process of developing a user-defined pattern follows these basic steps:

- The application developer works with the *pattern author* to develop an *exemplar* pattern. An exemplar pattern contains message flows, message maps, ESQL modules, Java classes for JavaCompute nodes, XML files, and any other components that are needed to define the base content for the pattern. The exemplar pattern should reflect good practices for the organization, since it serves as the basis for any instantiations of pattern instances.
- The pattern author reviews the exemplar pattern and determines which elements of the exemplar should be customizable by a user of the pattern. The developer and pattern author typically determine jointly which of the parameters should be customizable and how the pattern user should be able to configure them. The pattern author creates a pattern authoring project in the Integration Toolkit. After defining all the pattern plug-ins, the pattern author generates the pattern plug-ins. The Integration Toolkit creates a project file that contains the plug-in content.
- The integration developer and the pattern author test the pattern plug-in file. This testing allows the author to see the same image that a user sees after the pattern plug-in is installed on the user workstation.

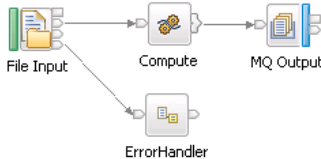
- After the author is satisfied with the user-defined pattern, the author creates a *pattern* archive. The archive contains the pattern plug-ins and all the elements of the user-defined pattern.
- The pattern archive is made available to users. The archive can be sent to a user, or posted on a shared file system or on a website (such as a pattern community website) for downloading.

WebSphere Education

IBM

Creating the pattern exemplar (1 of 4)

- The *pattern exemplar* is the message flow and related components that serve as the foundation of the user-defined pattern
- Developed as an Integration Bus application



```
graph LR; FI[File Input] --> C[Compute]; FI --> EH[ErrorHandler]; C --> MQ[MQ Output];
```

The diagram illustrates a message flow for a pattern exemplar. It starts with a 'File Input' node, which branches into two paths: one leading to a 'Compute' node and another leading to an 'ErrorHandler' node. The 'Compute' node then connects to an 'MQ Output' node. The 'ErrorHandler' node is positioned below the 'Compute' node, indicating a subflow for error handling.


© Copyright IBM Corporation 2013, 2015

Creating the pattern exemplar (1 of 4)

The first step in developing a user-defined pattern is to create the message flow and related components to serve as a foundation for the pattern. The exemplar can be as simple or as complex as needed for your specific application but it should be a proven solution.

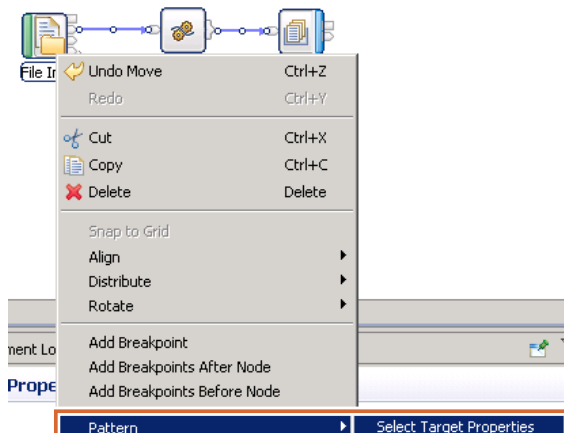
The Integration Bus developer develops the exemplar, usually with the pattern author, so that the expectations of the user-defined pattern are clearly understood.

In this example, the pattern exemplar consists of a File Input node, a Compute node, an MQ Output node, and an error handling subflow.

WebSphere Education 

Creating the pattern exemplar (2 of 4)

- Select the elements that you want the pattern user to configure
 - Draw box around selected elements, right-click, and then click **Pattern > Select Target Properties**




© Copyright IBM Corporation 2013, 2015

Creating the pattern exemplar (2 of 4)

The next step is to define the message flow properties that the pattern user can configure at pattern instantiation.

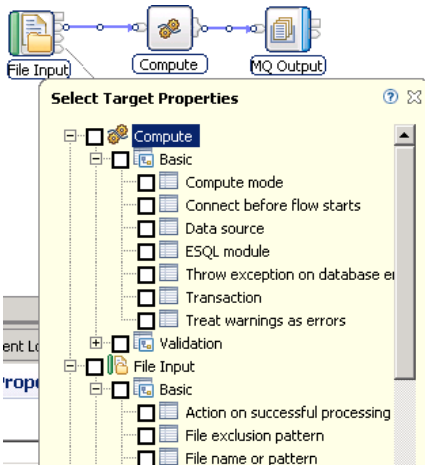
First, select the nodes that you want to make user-configurable by dragging a box around the nodes. The node outlines and wires turn blue.

Then, right-click a node or wire and click **Pattern > Select Target Properties**.

WebSphere Education 

Creating the pattern exemplar (3 of 4)

- Select the properties that the pattern user can configure



© Copyright IBM Corporation 2013, 2015

Creating the pattern exemplar (3 of 4)

When you click **Select Target Properties**, a dialog box is displayed. It contains a list of all the configurable properties for the selected objects.

Select the properties that you want the pattern user to be able to alter when the pattern is instantiated. When you are finished, close the dialog box by clicking the **X** in the upper right corner.

WebSphere Education

IBM

Creating the pattern exemplar (4 of 4)

- Elements in pattern exemplar message flow that user can customize are flagged

```
graph LR; FI[File Input] --> C[Compute]; C --> MO[MQ Output]; C --> EH[ErrorHandler];
```

© Copyright IBM Corporation 2013, 2015

Creating the pattern exemplar (4 of 4)

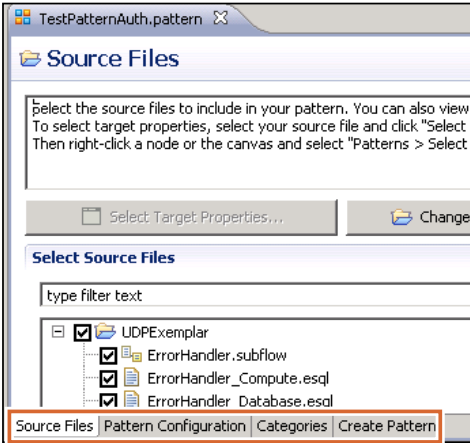
After you finish defining the configurable properties, the exemplar is updated with a “decorator” above each node in the exemplar message flow that contains user-configurable properties.

WebSphere Education

IBM

Pattern Authoring project

1. Create a Pattern Authoring project to contain the exemplar: **File > New > Pattern Authoring Project**
2. Select the artifacts to include in the project
 - At minimum, include the exemplar message flow application
3. Pattern Authoring editor opens and shows the exemplar properties
4. Follow tabs at bottom of Pattern Authoring editor to configure and create the pattern



© Copyright IBM Corporation 2013, 2015

Pattern Authoring project

After the exemplar is created, the work of authoring the user-defined pattern begins. You start by creating a Pattern Authoring Project in the Integration Toolkit.

The tabs in the Pattern authoring editor guide you through pattern development.

WebSphere Education

IBM

Selecting the source files

1. Click the **Source Files** tab
2. Select the source files to include in the user-defined pattern

Select Source Files

type filter text

☒

UDPExemplar

- ☒

ErrorHandler.subflow
- ☒

ErrorHandler_Compute.esql
- ☒

ErrorHandler_Database.esql
- ☒

UDPE_flow.msgflow
- ☒

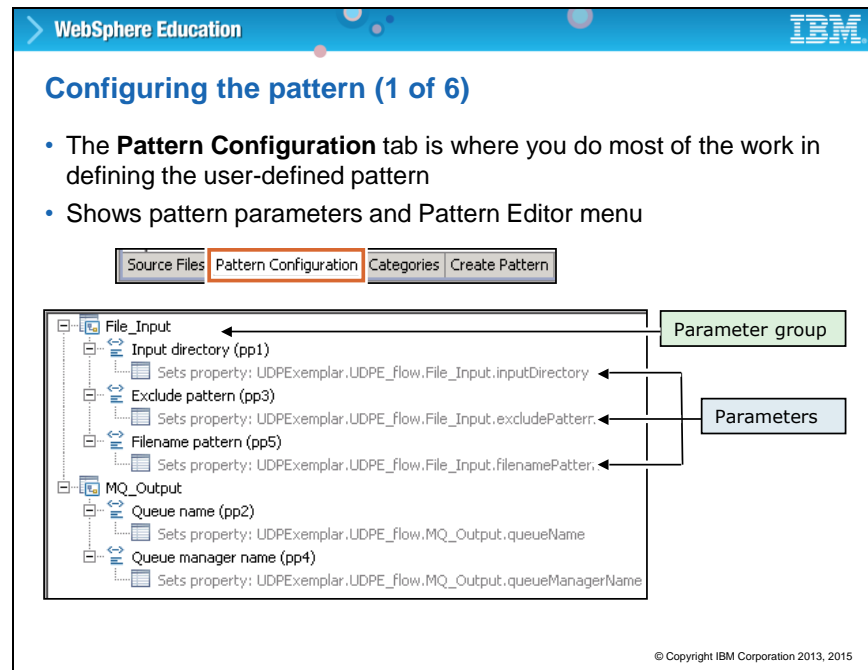
UDPE_flow_Compute.esql

Source Files | Pattern Configuration | Categories

© Copyright IBM Corporation 2013, 2015

Selecting the source files

In the first tab section of the Pattern authoring editor, you select the source files from the exemplar that you want to include in the final user-defined pattern. By default, all components are included. Before you elect to omit one or more files, consider the effect on the final project (for example, whether you want to include ESQL code for the pattern user).



Configuring the pattern (1 of 6)

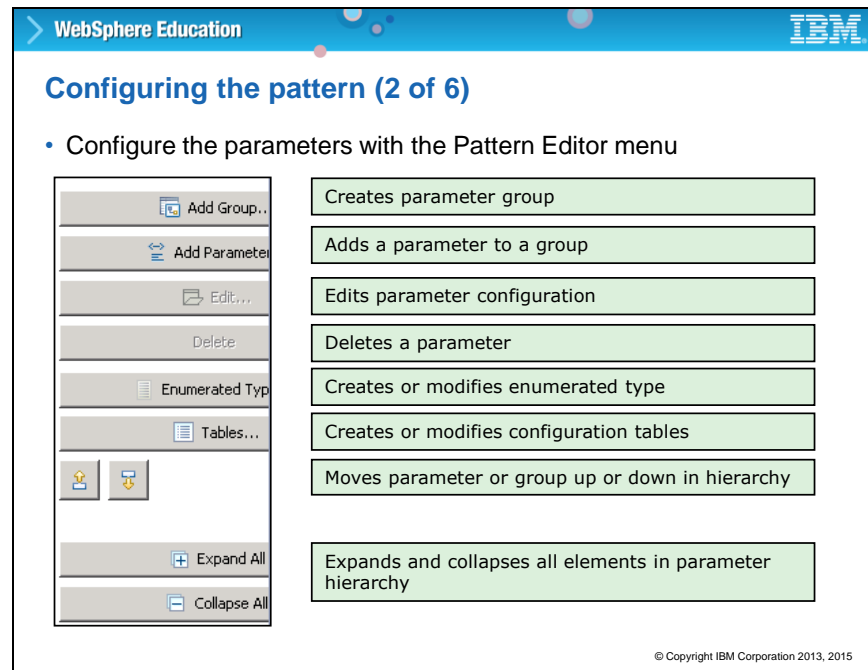
The next step of pattern authoring is typically where you spend most of your time when developing a user-defined pattern: pattern configuration.

In the pattern authoring project, the pattern author builds the user interface with which the pattern user interacts. The user interface specifies the configurable pattern parameters, and documents the parameters so that the user knows how to set the appropriate values at pattern instantiation time. Parameters can be defined as mandatory, optional, or hidden, and can be presented to the user in a number of formats such as a list, radio button, or a check box. The author can create complex transforms on parameters, including checking the content of parameter fields and doing conditional processing that is based on those values.

The parameters can be grouped in to logical groupings so that they are presented together to the user. The pattern author defines the field labels and field prompts that are available to the user, with help text to guide the user. The pattern author also defines the pattern name, pattern category, and help text that is shown when the user installs the pattern into the Patterns Explorer. The author can also define the version number and the identifying information for the pattern. All of these artifacts become the pattern plug-in.

Patterns are associated with groups. By default, these groups are the nodes with which they are associated in the pattern exemplar. For example, the screen capture shows that **File_Input** is a parameter group that contains the **Input_directory**, **Exclude_pattern**, and **Filename_pattern**

parameters. These parameters correspond to the configurable parameters that were selected for the FileInput node in the pattern exemplar. You can create, delete, and modify parameters and parameter groups in the pattern authoring editor. You can also associate (and disassociate) parameters with groups. You can also move parameters and groups within the displayed hierarchy.



Configuring the pattern (2 of 6)

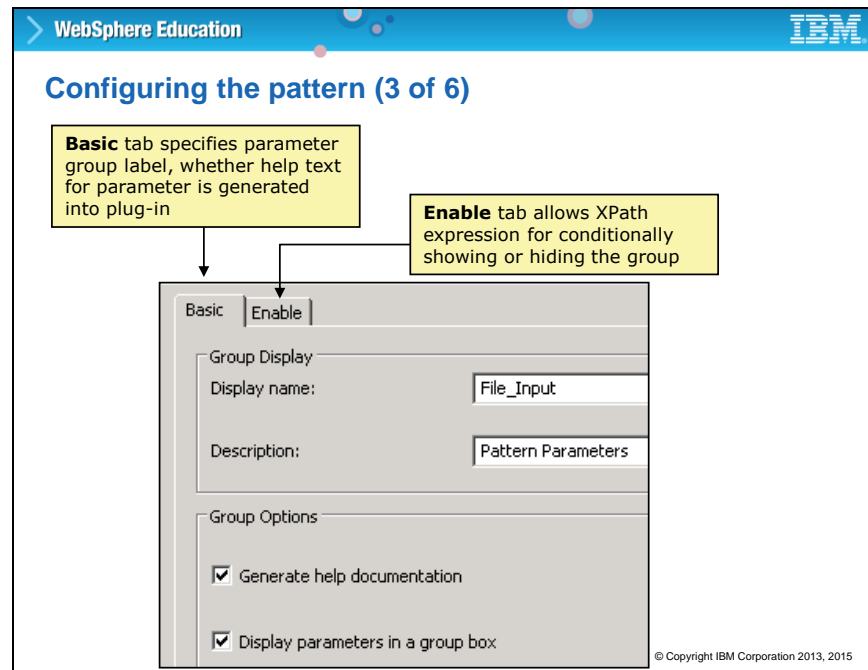
This slide shows the Pattern editor menu, which is displayed in the right side of the Pattern editor window.

The pattern author uses the Pattern editor menu to configure the pattern parameters. As you see on this slide, you can create parameters and parameter groups, edit parameter configurations, and move parameters and groups within the hierarchy. You can also define enumerate types and tables to use in parameter configuration.

These functions (and others) are also available by right-clicking a parameter or group.

As an option, you can define a pattern parameter as an enumerated type or a table.

- An **enumerated type** is a set of values that a parameter accepts. It consists of a type name and a list of display names and values.
- A **table** contains a collection of rows that the pattern user can configure. Each row in the table contains one or more columns of data.



Configuring the pattern (3 of 6)

When you edit a parameter group, two tabs are displayed.

When you select the **Basic** tab, you can edit the characteristics of the parameter group, including the name and description. You can also control whether the help documentation for the parameter group is included with the pattern documentation.

On the **Enable** tab, you can conditionally display or hide the parameter group, depending on the result of an XPath expression. You specify an XPath expression in the expression editor. If the expression evaluates to true, the pattern group is enabled and is shown to the pattern user. If the expression evaluates to false, the pattern group is disabled and the pattern user cannot see it. You can select the value of any other parameters that are defined in the pattern project to use in the XPath expression. This conditional processing capability is a flexible way to control the appearance of the pattern when the pattern user instantiates it.

WebSphere Education IBM

Configuring the pattern (4 of 6)

Basic Editor Transform Enable

Parameter Display
Display name:

Parameter Options
☐ Hide the parameter Select this option to hide the parameter. Use an XPath expression to specify the parameter to hide.
☒ Mandatory parameter Select this option if the parameter is mandatory. Mandatory parameters also display a red border.
 Field prompt:


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

© Copyright IBM Corporation 2013, 2015

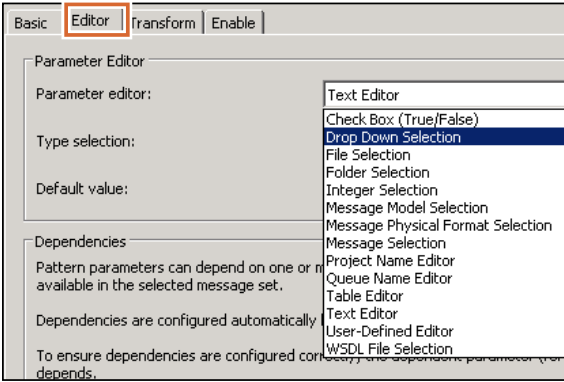
Configuring the pattern (4 of 6)

When you select a parameter to edit, four tabs are presented in the editor: **Basic**, **Editor**, **Transform**, and **Enable**.

On the **Basic** tab, you set parameter attributes such as the display name, whether the parameter is mandatory, and whether the parameter is hidden. You also set the field prompt, which is displayed in the parameter field itself at instantiation time. Also, you can create help text for the parameter by writing HTML. When you format the help text, a secondary window shows you how the help text is rendered to the pattern user.

WebSphere Education 

Configuring the pattern (5 of 6)



- **Editor tab:**
 - Sets the type of parameter and the control widget that the user uses to alter the parameter
 - Selects the most appropriate type automatically, but you can override
 - Specifies any dependency that this parameter has on any others

© Copyright IBM Corporation 2013, 2015

Configuring the pattern (5 of 6)

The **Editor** tab is where you control the specifics of the user interface for the parameter as it is shown to the pattern user. If the parameter type allows it, these attributes include the type of editor that is used to modify the field and provide an optional default value for the parameter.


The Parameter editor is based on the parameter type. In most cases, the Pattern authoring editor automatically selects the most appropriate parameter editor, depending on the parameter type, but you can override this value.

A number of parameter editors are available. For example, you can define a check box option, drop-down list, a file selection menu, or text editor.

Most of these editors contain configurable properties to assist the pattern user. For example, if you choose the **File Selection** editor, you are prompted to provide a list of file extensions that are displayed in the field prompt for the pattern user.

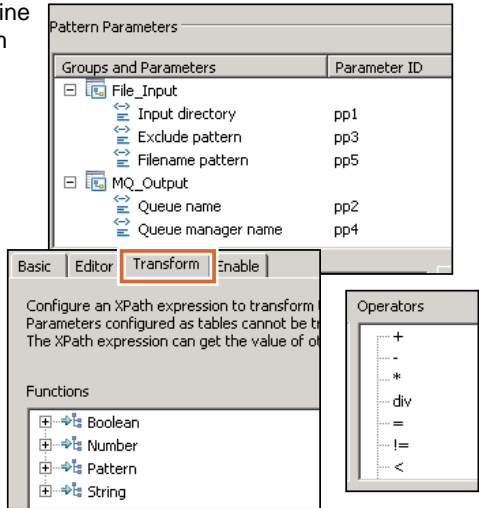
You can also create user-defined editors that are specific to the parameter type.

On the **Editor** tab, you can also view dependencies between parameters. For example, the editor that is used for a parameter can depend on the editor that is used for a second parameter, or on its values. The pattern authoring editor manages these dependencies for you, but the dependencies are listed here for your reference.

WebSphere Education 

Configuring the pattern (6 of 6)

- Use the **Transform** tab to define XPath expression to transform the value of the parameter
- Query the value of other parameters by using `getValue()` expression



| Groups and Parameters | Parameter ID |
|-----------------------|--------------|
| File_Input | |
| Input directory | pp1 |
| Exclude pattern | pp3 |
| Filename pattern | pp5 |
| MQ_Output | |
| Queue name | pp2 |
| Queue manager name | pp4 |

Basic | Editor | **Transform** | Enable

Configure an XPath expression to transform the value of the parameter. Parameters configured as tables cannot be transformed. The XPath expression can get the value of other parameters using the `getValue()` expression.

Functions

- Boolean
- Number
- Pattern
- String

Operators

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

© Copyright IBM Corporation 2013, 2015


Configuring the pattern (6 of 6)

You use the **Transform** tab to define expressions to transform the value of a parameter. When the pattern user creates the pattern instance by clicking **Generate** in the Pattern Explorer, the XPath expression is evaluated and the transform occurs.

The transform can evaluate the value of other parameters by using the `getValue()` XPath expression. An XPath expression builder is included on this tab to help you construct the appropriate XPath expressions and to evaluate them for testing.

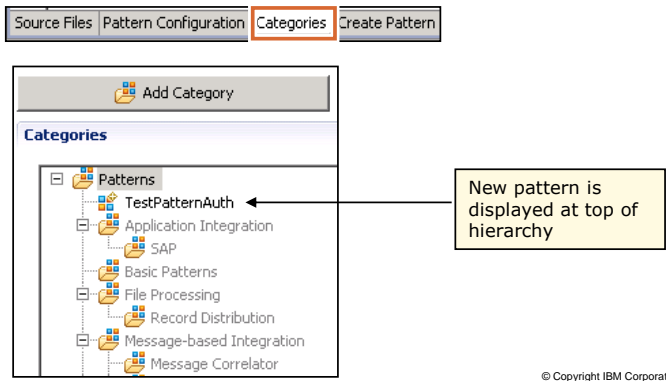
If you are using tables as part of the parameter configuration, you use Java code instead of XPath expressions to evaluate the table values.

You use the **Enable** tab to create an XPath expression that controls whether the parameter is visible to the pattern user in the Pattern Instance editor.

WebSphere Education 

Setting the Patterns Explorer category

- Use the **Categories** tab to set the location in the Patterns Explorer for the new pattern
 - Drag to location you want
 - Optionally, create a category for pattern



Source Files | Pattern Configuration | **Categories** | Create Pattern

Add Category

Categories

Patterns

- TestPatternAuth
- Application Integration
 - SAP
- Basic Patterns
- File Processing
 - Record Distribution
- Message-based Integration
 - Message Correlator

New pattern is displayed at top of hierarchy

© Copyright IBM Corporation 2013, 2015


Setting the Patterns Explorer category

Before you generate the pattern plug-ins, you must define where the new pattern is displayed in the Patterns Explorer in the Integration Toolkit.

First, select the **Categories** tab, and then drag the new pattern name to the location in the pattern hierarchy that you want.

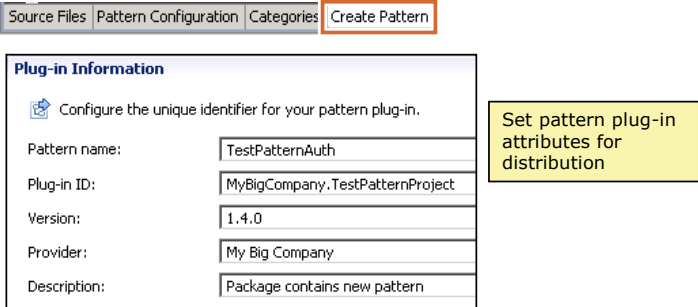
You can also create a category to contain the pattern by clicking **Add Category**. You can add the category at any location in the hierarchy.

Normally you set the pattern location after you customize all the parameters and parameter groups, but you can do it any time before you generate the pattern plug-ins.

WebSphere Education 


Creating the pattern (1 of 2)

- Use the **Create Pattern** tab to:
 - Create a file that contains the pattern plug-ins
 - Test the pattern, and debugging if necessary
 - Create the pattern archive that contains the distributable pattern package



Source Files | Pattern Configuration | Categories | **Create Pattern**

Plug-in Information

 Configure the unique identifier for your pattern plug-in.

| | |
|---------------|---------------------------------|
| Pattern name: | TestPatternAuth |
| Plug-in ID: | MyBigCompany.TestPatternProject |
| Version: | 1.4.0 |
| Provider: | My Big Company |
| Description: | Package contains new pattern |

Set pattern plug-in attributes for distribution

© Copyright IBM Corporation 2013, 2015

Creating the pattern (1 of 2)

You do the final steps of creating a user-defined pattern in the **Create Pattern** tab. These steps include:

- Setting the attributes for the pattern plug-ins file
- Generating a file that contains the pattern plug-ins
- Testing and debugging the pattern
- Creating the pattern archive file. This file contains the user-defined pattern in distributable form.

As you saw with the other aspects of pattern development, the pattern authoring editor guides you through these last steps by presenting a window that contains controls for you to use.

On the **Create Pattern** tab, you set attributes for the plug-ins file. These attributes are packaged with the plug-ins themselves. The information that you provide on this page typically follows your development standards for nomenclature, standards, and version numbers.

WebSphere Education

IBM

Creating the pattern (2 of 2)

- Before testing the pattern, you must create the pattern plug-ins
- Click **Create Pattern Plug-ins**

Create Pattern Plug-ins

- Can be a resource-intensive process, depending on the size and complexity of the pattern project

User Operation is Waiting

The user operation is waiting for "Building workspace" to complete.

Java annotation index: update

Waiting for Java search

Validating links

Building workspace (Waiting)

© Copyright IBM Corporation 2013, 2015

Creating the pattern (2 of 2)

Before you can test a pattern, you must generate a file that contains the pattern plug-ins by clicking **Create Pattern Plug-Ins** on the **Create Pattern** tab.

Depending on the size and the complexity of the pattern project, this operation can take several minutes and use much of the workstation processor power.

The file that contains the plug-ins is made available for testing.


WebSphere Education

IBM

Testing the pattern

- Testing a new pattern is an essential part of developing it
- When testing, the pattern author sees what the pattern user sees
- Test from the perspective of the user of the new pattern
 - Are the correct parameters included?
 - Do the prompts make sense?
 - Is the help text meaningful?
 - Does the generated message flow work as intended?

1. Click **Test Pattern**
2. A second copy of the Integration Toolkit opens. Specify a different workspace for testing
3. From the **Quick Starts** pane, click **Start from Patterns**

A rectangular button with a green play icon and the text "Test Pattern..."

© Copyright IBM Corporation 2013, 2015

Testing the pattern

A critical aspect of creating a user-defined pattern is testing it. When you test a pattern, a second copy of the Integration Toolkit opens. It includes the pattern plug-in that you created in the previous step.

During testing, the author should do the same as a user from selecting the pattern from the Patterns Explorer, through customizing the pattern parameters, to generating the pattern instance. This testing allows the author to determine whether any part of using the process must be modified.

Testing the pattern creates a workspace and shows the Patterns Explorer that includes the new pattern. You cannot use the same workspace that is open for the Pattern authoring editor in the original instance of the Toolkit.


Open the Patterns Explorer from the **Quick Start** pane by clicking **Start from Patterns**. The new pattern is displayed in the Patterns Explorer in the location you set in the previous steps. Select the pattern and begin testing.

As you test, consider yourself to be the pattern user. Some considerations for testing include:

- Are the correct parameters available for the user to configure?
- Do the configuration steps make sense, and are they presented in a logical progression?
- Are the correct editors used?

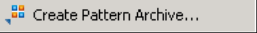
- Are the default values appropriate?
- If parameters and parameter groups were set to appear conditionally, do they appear (or not appear) as expected?
- Is the documentation for the pattern and the parameters correct?
- Does the generated message flow work as intended?

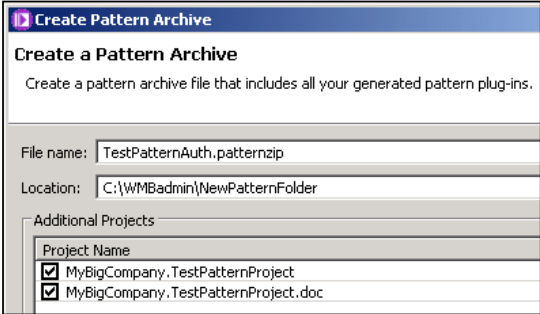
When you are finished testing the pattern plug-in, close the second copy of the Integration Toolkit.

WebSphere Education


Creating the pattern archive

Pattern archive is the compressed file that is used to distribute the new pattern to users

1. Click **Create Pattern Archive** 
2. You are prompted for the components to include in the pattern archive and the archive file name and location



Create Pattern Archive

Create a Pattern Archive

Create a pattern archive file that includes all your generated pattern plug-ins.

File name: TestPatternAuth.patternzip

Location: C:\WMBadmin\NewPatternFolder

Additional Projects

| Project Name |
|---|
| <input checked="" type="checkbox"/> MyBigCompany.TestPatternProject |
| <input checked="" type="checkbox"/> MyBigCompany.TestPatternProject.doc |

© Copyright IBM Corporation 2013, 2015

Creating the pattern archive

After you complete all development and testing for the new pattern, you can package it for distribution by creating a pattern archive. A pattern archive is a compressed file that contains all the artifacts that are needed to deploy the pattern to other Integration Toolkit installations.

To create the pattern archive, click **Create Pattern Archive** on the **Create Pattern** tab. You are prompted for the name and location for the archive file, and the components to include in it.

Like the generation process for the pattern plug-ins, creating the pattern archive can be resource-intensive and can take an extended amount of time to complete.

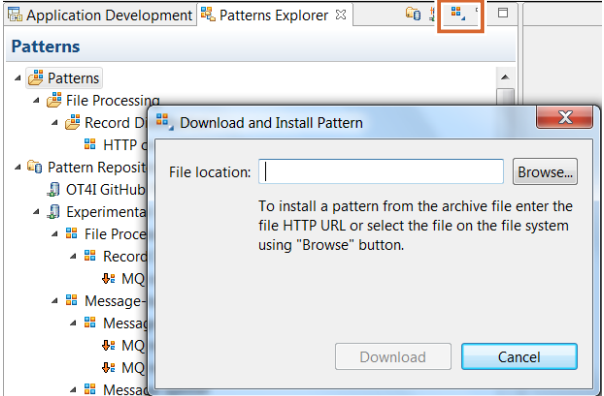
After the file is generated, you can distribute it according to the policies of your organization.

WebSphere Education

IBM

Installing the user-defined pattern

- A user can download and install the new pattern from the **Patterns Explorer** in the IBM Integration Toolkit
- Click the **Download and install** icon and then specify the download location



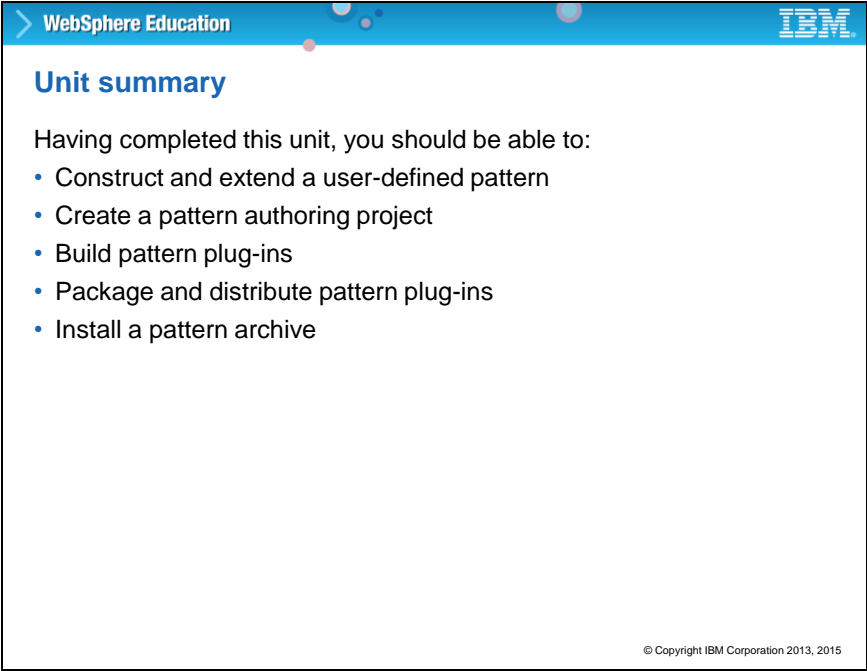
The screenshot shows the 'Patterns Explorer' window in the IBM Integration Toolkit. A red box highlights the 'Download and Install' icon in the toolbar. A dialog box titled 'Download and Install Pattern' is open, featuring a 'File location:' text field, a 'Browse...' button, and instructional text: 'To install a pattern from the archive file enter the file HTTP URL or select the file on the file system using "Browse" button.' At the bottom of the dialog are 'Download' and 'Cancel' buttons. The background shows a tree view of pattern categories like 'File Processing', 'Record Driven', and 'HTTP'.

© Copyright IBM Corporation 2013, 2015


Installing the user-defined pattern

To install the user-defined pattern in the Integration Toolkit, you use the Patterns Explorer.

Click **Download**, and then select the location of the pattern archive (.patternzip) file. When the file is located, the Integration Toolkit automatically extracts the components and adds the pattern to the Patterns Explorer.



The slide features a blue header bar with the text 'WebSphere Education' on the left and the IBM logo on the right. Below the header, the title 'Unit summary' is displayed in blue. The main content area contains a paragraph followed by a bulleted list of five items. At the bottom right, there is a small copyright notice.

> WebSphere Education 

Unit summary

Having completed this unit, you should be able to:

- Construct and extend a user-defined pattern
- Create a pattern authoring project
- Build pattern plug-ins
- Package and distribute pattern plug-ins
- Install a pattern archive

© Copyright IBM Corporation 2013, 2015

Unit summary

This unit described how to create and test user-defined message flow patterns in the IBM Integration Toolkit.

Having completed this unit, you should be able to:

- Construct and extend a user-defined pattern
- Create a pattern authoring project
- Build pattern plug-ins
- Package and distribute pattern plug-ins
- Install a pattern archive