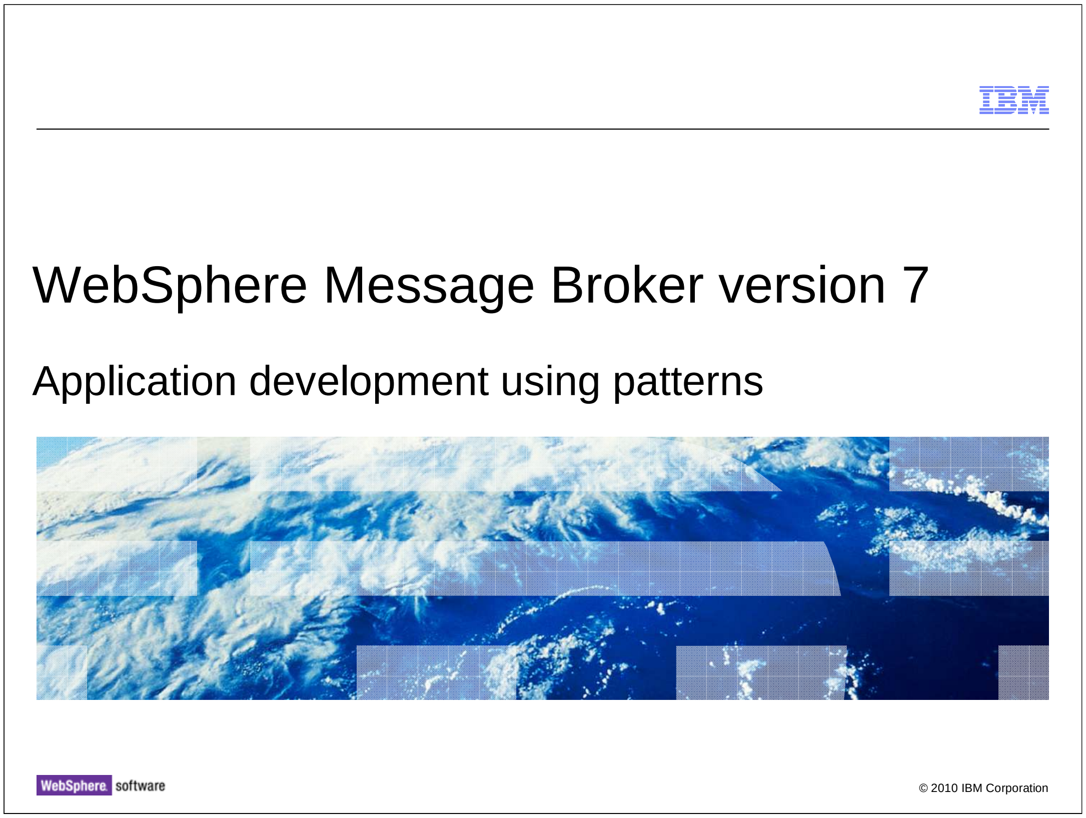
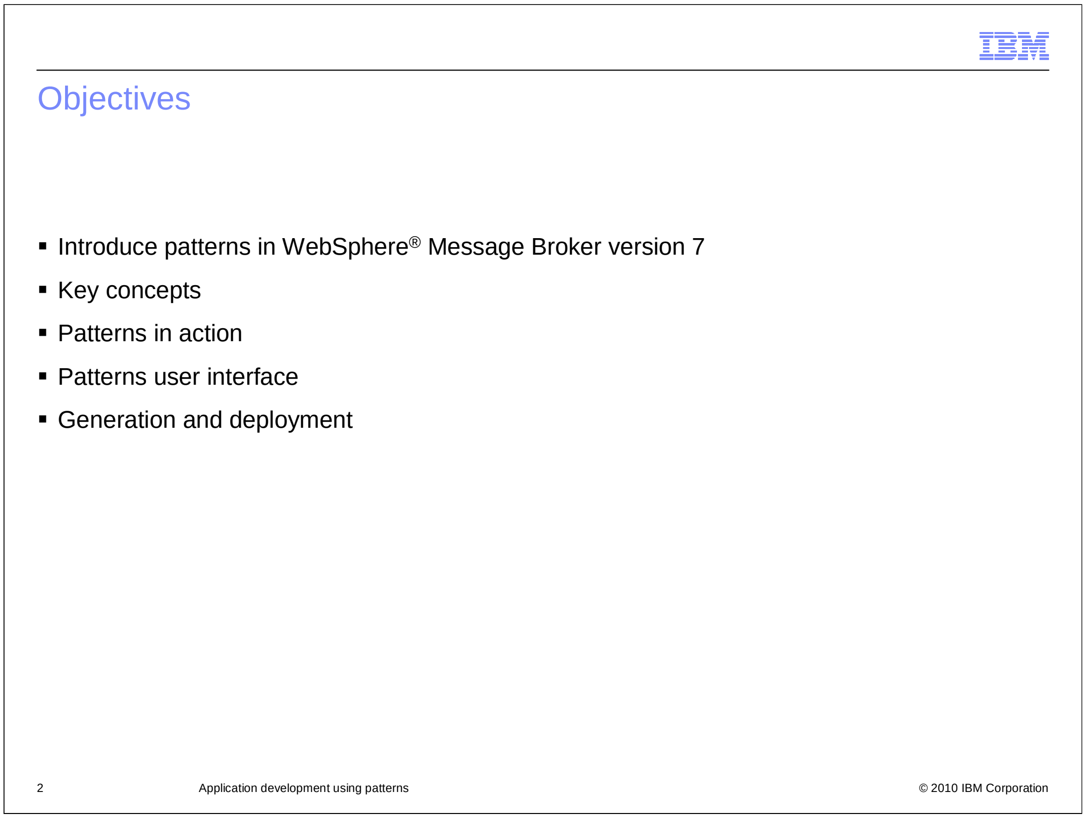


WebSphere Message Broker version 7

Application development using patterns

WebSphere software

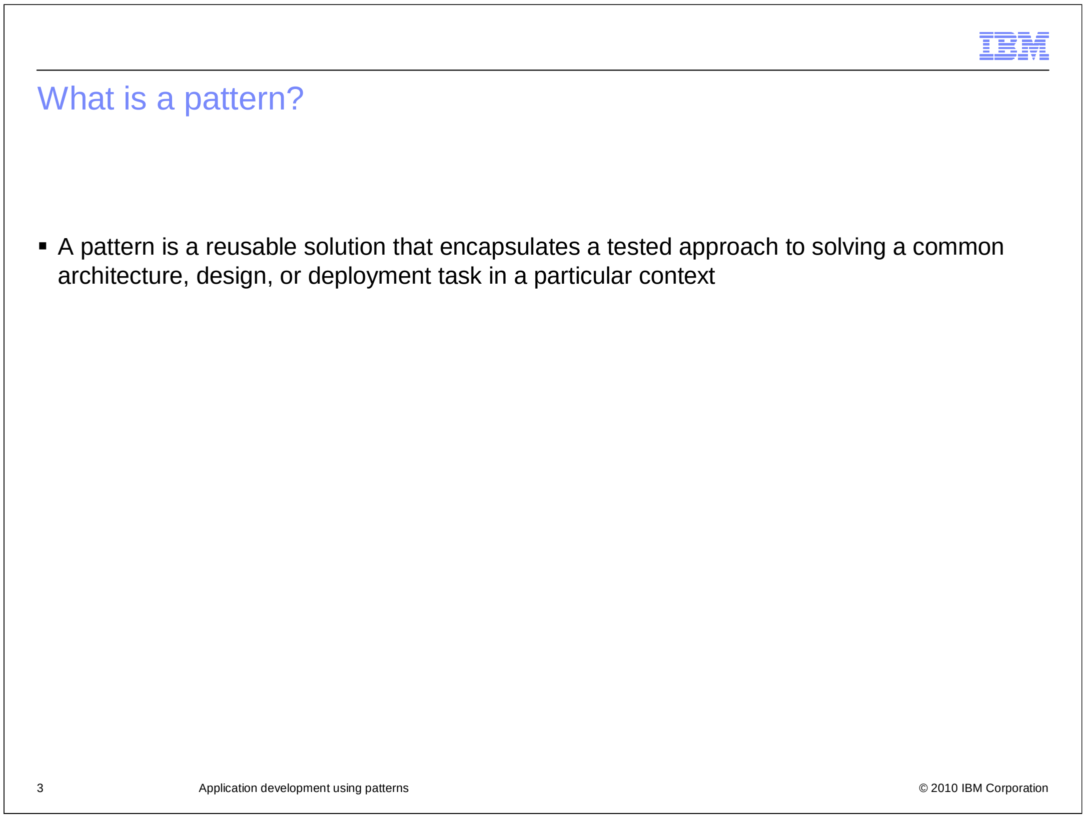
This session will describe the new tools provided in Message Broker version 7 for application development using patterns.



Objectives

- Introduce patterns in WebSphere® Message Broker version 7
- Key concepts
- Patterns in action
- Patterns user interface
- Generation and deployment

Pattern-based application development is an application development style that is gaining rapid acceptance in the IT industry. Pattern-based development is intended to provide a style of application development that is rapid, and generates new applications that are consistent, and have the same basic structure when they have been fully tested and implemented. This session will describe how patterns have been implemented in Message Broker, and how they are presented to you. It will show you how to create a pattern instance, and how to generate a new application based on your chosen pattern instance.

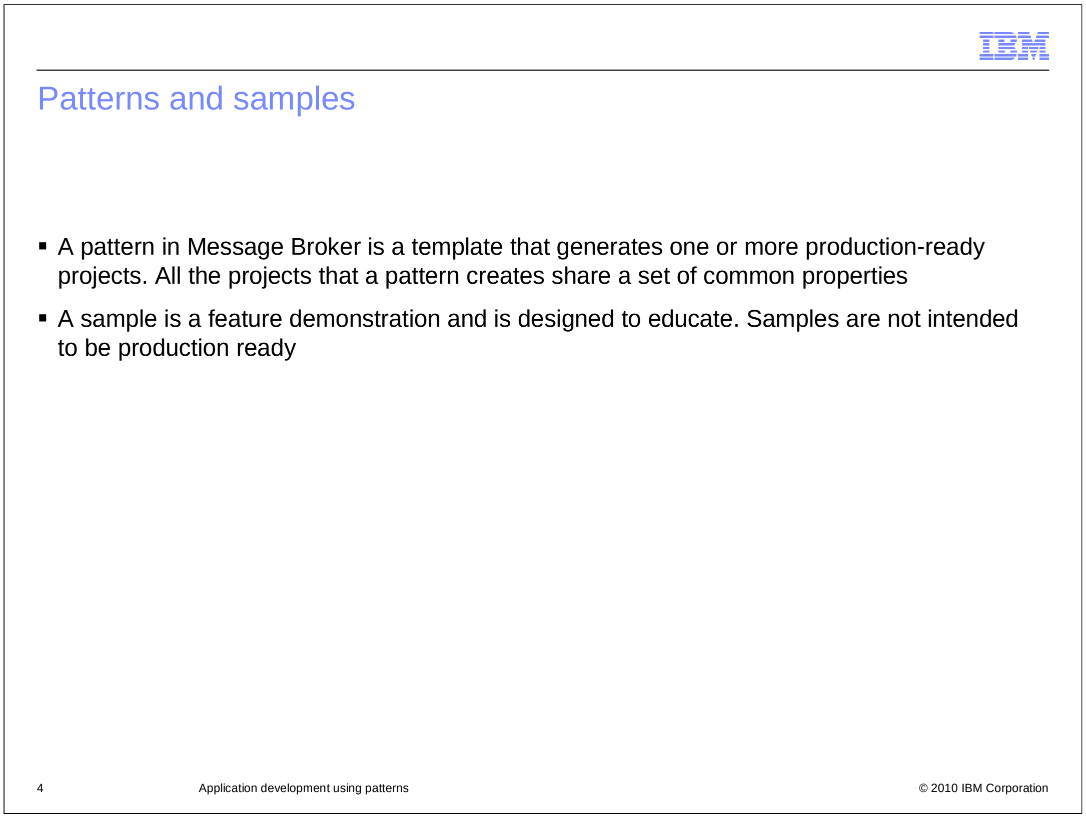


What is a pattern?

- A pattern is a reusable solution that encapsulates a tested approach to solving a common architecture, design, or deployment task in a particular context

A pattern is a pre-built template that can be used as the basis for a Message Broker application. As part of the generation process, you supply the parameters required for the new application. The pattern will then be used to build a set of application artifacts such as message flows and message sets. The resulting applications are designed to be production ready. However, of course, you will normally extend the generated applications to perform the specific tasks of your application.

The resulting message broker applications are called pattern instances, and all generated instances of the same pattern will have the same basic structure



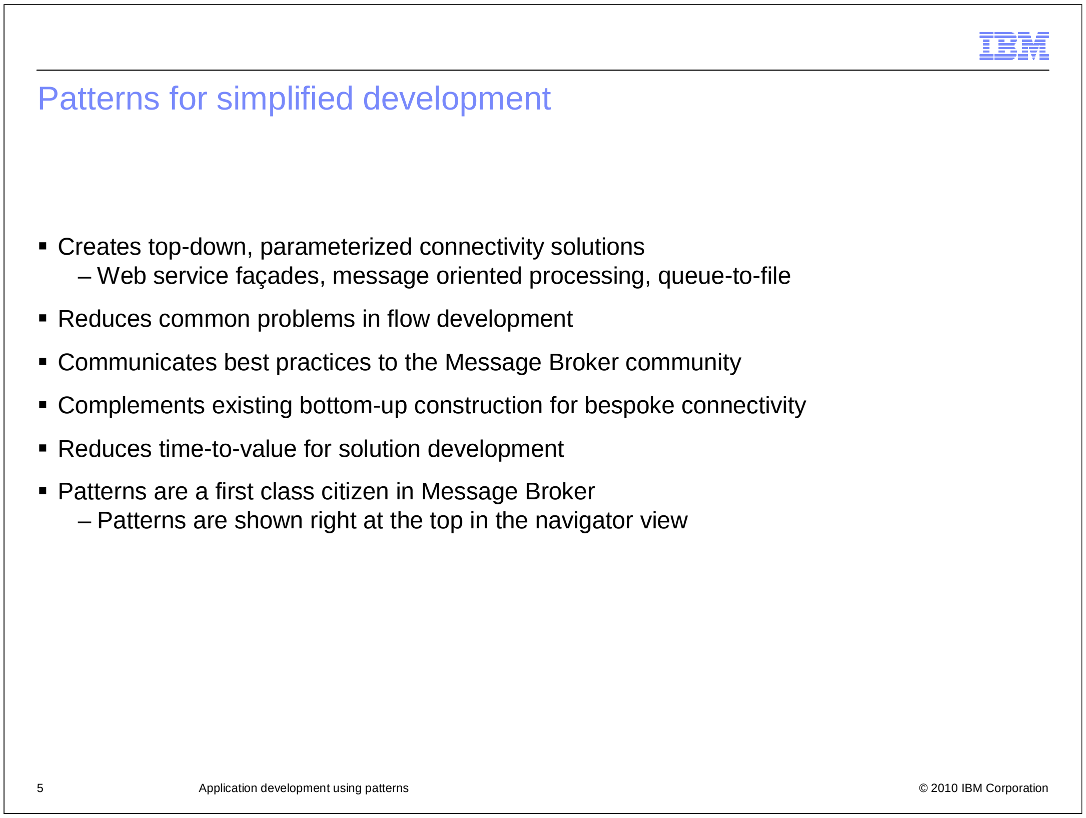
Patterns and samples

- A pattern in Message Broker is a template that generates one or more production-ready projects. All the projects that a pattern creates share a set of common properties
- A sample is a feature demonstration and is designed to educate. Samples are not intended to be production ready

4 Application development using patterns © 2010 IBM Corporation

It is important to understand the between the difference between a pattern and a sample. Message Broker version 7 contains a range of sample applications. These are provided to illustrate how to use certain key functions in the Message Broker product, and all new major application functions will have an accompanying sample. These are accessible from the samples gallery. Sample applications are illustrative only, and are not written to be production-ready.

Conversely, the supplied patterns contained within Message Broker version 7 are designed to generate production-ready applications. These applications exemplify best practice, and can be relied on to form a strong basis for extending with your own application-specific function.



Patterns for simplified development

- Creates top-down, parameterized connectivity solutions
  – Web service façades, message oriented processing, queue-to-file
- Reduces common problems in flow development
- Communicates best practices to the Message Broker community
- Complements existing bottom-up construction for bespoke connectivity
- Reduces time-to-value for solution development
- Patterns are a first class citizen in Message Broker
  – Patterns are shown right at the top in the navigator view

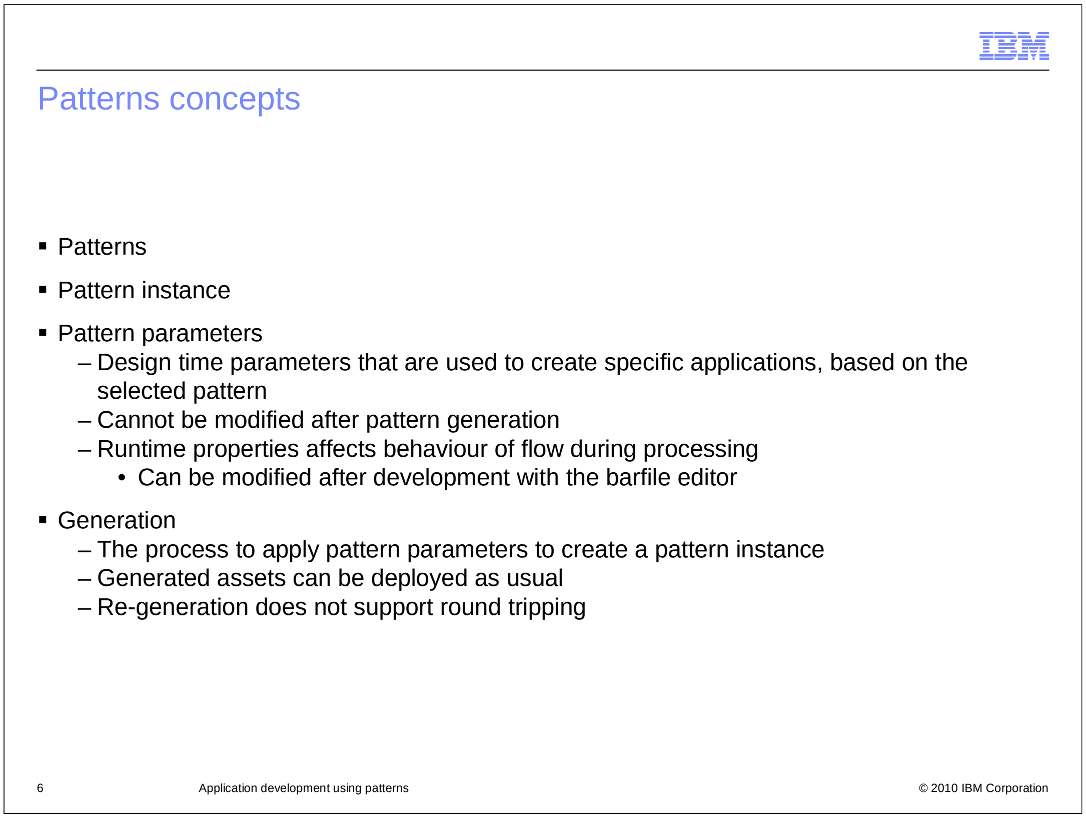
5 Application development using patterns © 2010 IBM Corporation

Patterns-based development has been introduced in Message Broker to address several requirements.

Many application designers want to create new applications based on the provision of a set of parameters, rather than creating the individual components of a messaging application. For example, an application that provides a Web service façade will typically follow the same structure each time it is implemented. Using a pattern-based approach enables these types of applications to be built in the same way, by using a set of parameters to define the specific application requirements.

Many application problems have a common cause. Eliminating the causes of such problems at the outset, and using the appropriate best practices for application development techniques will ensure greater productivity for your own applications.

The supplied patterns are shown in the Message Broker toolkit, by selecting the patterns navigator.



Patterns concepts

- Patterns
- Pattern instance
- Pattern parameters
  - Design time parameters that are used to create specific applications, based on the selected pattern
  - Cannot be modified after pattern generation
  - Runtime properties affects behaviour of flow during processing
    - Can be modified after development with the barfile editor
- Generation
  - The process to apply pattern parameters to create a pattern instance
  - Generated assets can be deployed as usual
  - Re-generation does not support round tripping

6 Application development using patterns © 2010 IBM Corporation

There are four primary concepts within patterns in Message Broker.

First, the set of supplied patterns is comprised of several application templates that are available in the Message Broker toolkit, using the patterns explorer.

Second, when you have chosen the appropriate pattern, you will create a pattern instance. This is a separate project in the workspace, and is shown in a separate area of the available projects. This area is entitled “Pattern Instances”.

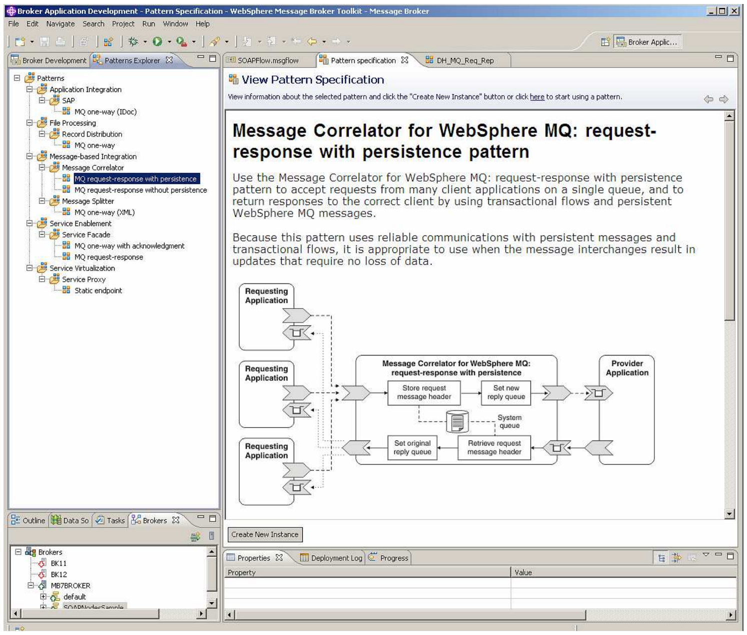
Thirdly, this pattern instance will be combined with several parameters. These parameters are used to specify your unique application requirements.

When you have provided all your parameter values, the generation process will create a new project in your toolkit workspace, in exactly the same way as a traditional Message Broker project. This project will contain message flows, message sets, and other artifacts, depending on the selected pattern, and your pattern parameters.

After the generation process, the remaining steps in the application development process can continue as normal. This will comprise the development of further application requirements, the construction of the barfile, and the deployment of the barfile along with the necessary testing steps. As with normal application development, the runtime parameters in the barfile can be overridden at deployment time.

After the application has been generated, the application can not be re-generated. The application must now be managed as a normal Message Broker application.

Patterns explorer



Broker Application Development - Pattern Specification - WebSphere Message Broker Toolkit - Message Broker

Broker Development | Patterns Explorer

- Patterns
  - Application Integration
    - SAP
      - MQ one-way (IDoc)
  - File Processing
    - Record Distribution
      - MQ one-way
  - Message-based Integration
    - Message Correlator
      - MQ request-response with persistence
      - MQ request-response without persistence
    - Message Splitter
      - MQ one-way (XML)
  - Service Enablement
    - Service Facade
      - MQ one-way with acknowledgment
      - MQ request-response
  - Service Virtualization
    - Service Proxy
      - Static endpoint

SOAPFlow.msgflow | Pattern specification | DH_MQ_Req_Rep

View Pattern Specification

View information about the selected pattern and click the "Create New Instance" button or click here to start using a pattern.

Message Correlator for WebSphere MQ: request-response with persistence pattern

Use the Message Correlator for WebSphere MQ: request-response with persistence pattern to accept requests from many client applications on a single queue, and to return responses to the correct client by using transactional flows and persistent WebSphere MQ messages.

Because this pattern uses reliable communications with persistent messages and transactional flows, it is appropriate to use when the message interchanges result in updates that require no loss of data.

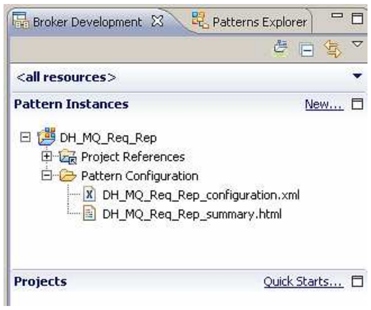
- Brokers
  - BK11
  - BK12
  - MB7BROKER
    - default
    - SOAPNodesSample

The supplied patterns are accessed by clicking on the Patterns Explorer tab in the project navigator. Message Broker version 7 has supplied eight patterns. Highlighting these, by clicking on them, will show the pattern specification on the right. A schematic diagram of the pattern is shown. Further text description of the pattern can be seen by scrolling down to the lower part of the window.

Patterns are grouped into categories. For example, there is a primary group called “Message-based integration”. This has sub-groups called “Message correlator” and “Message splitter”. At the lowest level are the patterns themselves. As shown here, the pattern called “MQ Request-Response with Persistence” has been selected.

You can create a pattern instance of this pattern by clicking “Create New Instance”.

Pattern instances in the Message Broker project navigator

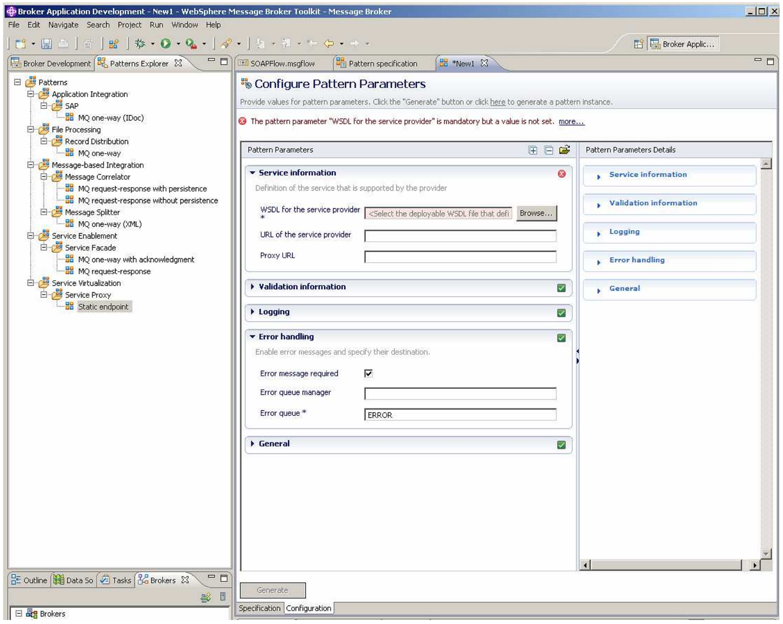


- User is directed towards a new entry point for development
- First class development mechanism
  - Augments other starting points:
    - WSDL, SCA, adapter
- Does not replace the existing bottom up development approach
  - Still valid to start from message flows, message sets …

When you have created a new pattern instance, you will see the resulting project files in the Message Broker navigator. The pattern instance project is placed at the top of the navigator list. It will be included in the defined working set, so selecting another working set will hide the created pattern instance files.

You can start application development using patterns in the same way as any other starting point. In particular, the set of quick starts now includes “Start from Patterns” as one of the primary starting points. Alternatively, you can continue to create a message flow and message set from scratch.

Configuring pattern parameters



Configure Pattern Parameters

Provide values for pattern parameters. Click the "Generate" button or click here to generate a pattern instance.

The pattern parameter "WSDL for the service provider" is mandatory but a value is not set. more...

Pattern Parameters

Service information

Definition of the service that is supported by the provider

WSDL for the service provider * <Select the deployable WSDL file that defi Browse...

URL of the service provider

Proxy URL

Validation information ✓

Logging ✓

Error handling ✓

Enable error messages and specify their destination.

Error message required ☑

Error queue manager

Error queue * ERROR

General ✓

Pattern Parameters Details

Service information

Validation information

Logging

Error handling

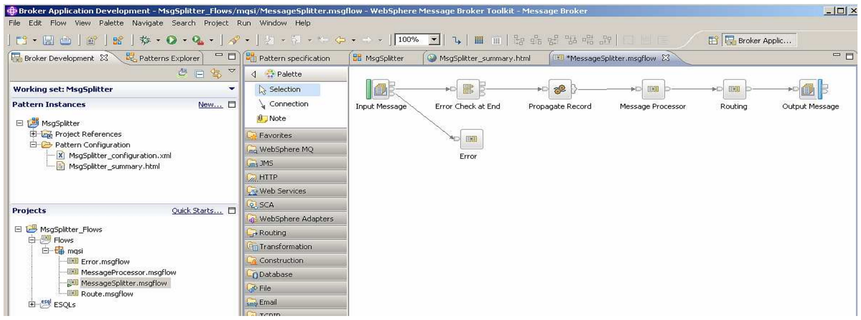
General

When you have created the required pattern instance, the “Configure Pattern Parameters” window is shown. The parameters are grouped into sections that contain logically related items. Most parameters are optional, and some patterns do not require the provision of any further parameters to enable the generation of the application. In the example shown here, the Services proxy pattern, the WSDL parameter is mandatory, but all other parameters can be left. These are indicated with a green tick. Note however, that each pattern generation process will use the provided parameters, so you should take care that applications will not use resources used by other applications.

When you have provided all the required parameters, click the Generate button to create the message flow and other artifacts.

Understanding the generated artifacts

- A working set is created, it includes only the current pattern
- The ***Projects*** view contains the generated projects
- The ***Patterns*** view contains the pattern instance projects
  - – Includes the configuration XML and summary page
- Patterns does not attempt to provide life cycle management



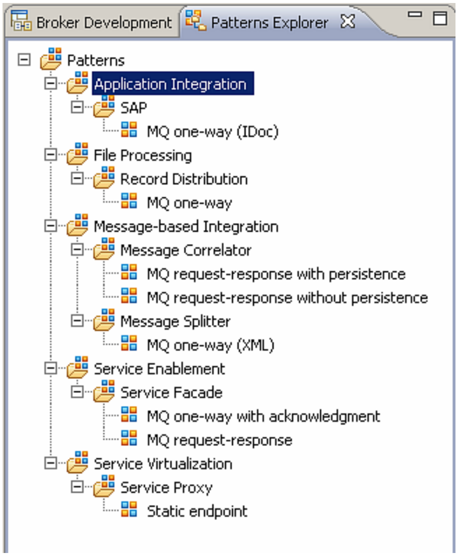
All generated message flows, message sets, and any other project artifact is included in the generated working set. In the example shown here, the Message Splitter working set has been created, and all generated artifacts are shown in this working set. This includes the pattern instance that was used to generate the message broker artifacts. The XML file contained in the pattern instance project contains all the parameters that you specified when you generated the project. If you open this XML file, the pattern parameters editor will open, and you can make further changes to your parameters.

The pattern instances project also contains a summary file, which is an HTML file containing a more readable view of the selected parameters.

If you need to re-generate the application artifacts, these will be over-written when you select the generate process. If you have made any changes to the generated application, these will be lost when the application is generated.

The pattern categories

- Application integration
- File processing
- Message-based integration
- Service enablement
- Service virtualization



- Patterns
  - Application Integration
    - SAP
      - MQ one-way (IDoc)
  - File Processing
    - Record Distribution
      - MQ one-way
  - Message-based Integration
    - Message Correlator
      - MQ request-response with persistence
      - MQ request-response without persistence
    - Message Splitter
      - MQ one-way (XML)
  - Service Enablement
    - Service Facade
      - MQ one-way with acknowledgment
      - MQ request-response
  - Service Virtualization
    - Service Proxy
      - Static endpoint

11 Application development using patterns © 2010 IBM Corporation

Message Broker version 7 has included eight patterns, within five major categories.

The application integration category contains one pattern, intended for connectivity to SAP. This pattern is intended for SAP systems which send an IDOC to Message Broker for further processing. The first message flow examines the type of IDOC, then routes the resulting message to the appropriate message flow for final processing.

The File Processing category contains patterns related to file processing. The supplied pattern takes an input file, and writes each fire record out as an MQ message.

The Message-based category has three MQ patterns.

The MQ request-response pattern accepts requests from many client applications on a single queue, and returns responses to the correct client. This is provided in two styles, with and without transactional flows and persistent MQ messages.

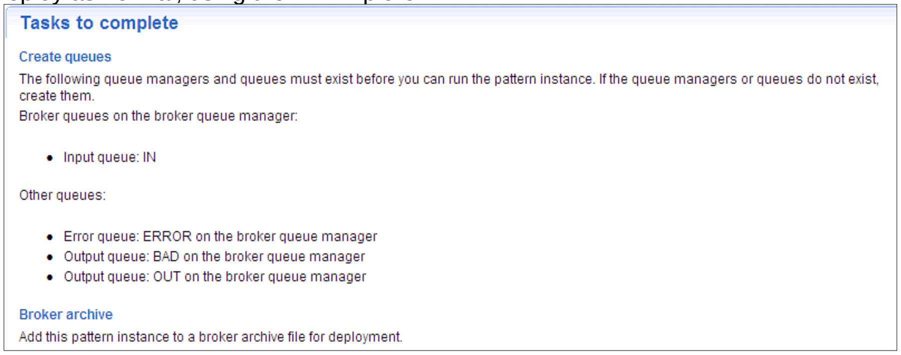
The MQ message splitter can be used to split a large XML message into smaller elements for processing by one or more targets by using transactional flows and persistent MQ messages.

The service enablement category provides a Web service façade for a variety of server applications. The supplied patterns provide a service façade for two MQ-based styles of application .

And finally, the service virtualization category is used to provide maximum decoupling between service requestors and service providers by introducing an additional level of indirection to the service invocation. This can be particularly useful in providing a management point for access control, request tracking, and auditing.

Pattern deployment

- Pattern instance can create additional runtime controls
  - For example, a user-defined property to control error logging or trace
  - These can be customized as normal
- Deployment
  - Build barfile in the normal way from generated assets
  - Deploy as normal, using the MB Explorer



Tasks to complete

Create queues

The following queue managers and queues must exist before you can run the pattern instance. If the queue managers or queues do not exist, create them.

Broker queues on the broker queue manager:

- Input queue: IN

Other queues:

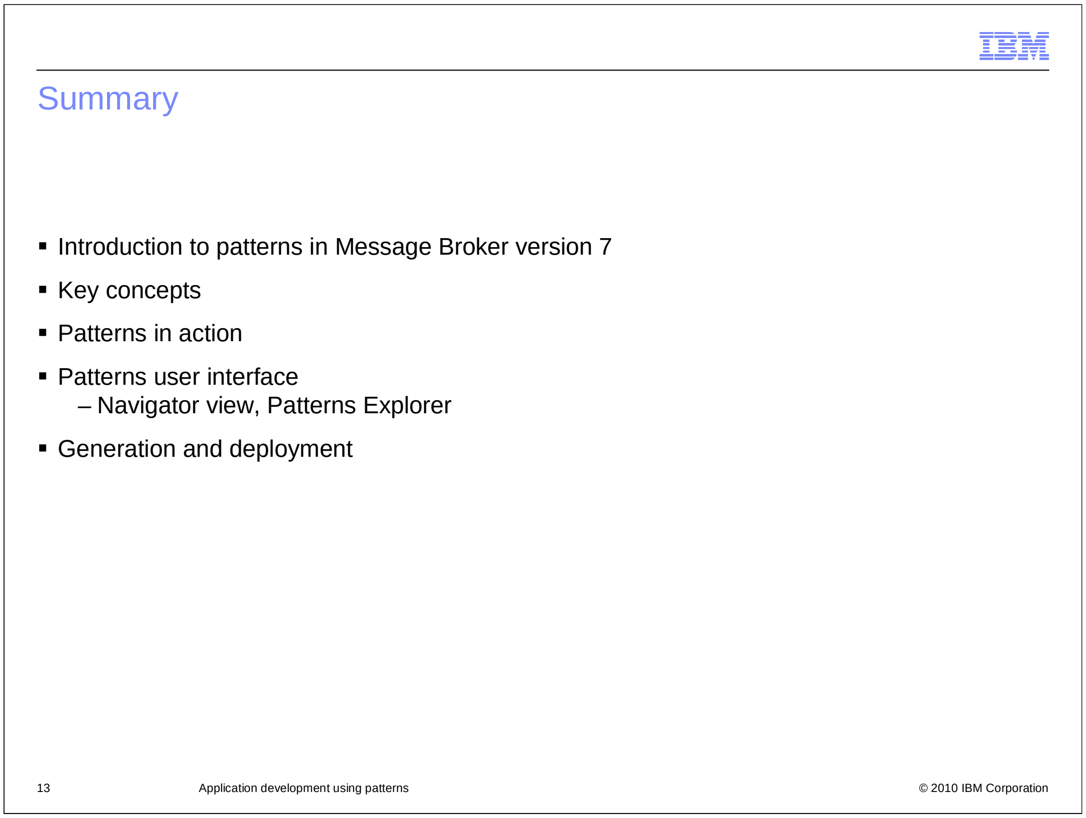
- Error queue: ERROR on the broker queue manager
- Output queue: BAD on the broker queue manager
- Output queue: OUT on the broker queue manager

Broker archive

Add this pattern instance to a broker archive file for deployment.

The deployment of a pattern-based application is exactly the same as a traditional application. The application is deployed using the barfile mechanism, in any of the typical styles.

In addition, the pattern generation process will include a description of the remaining tasks to complete the application. This is contained in the summary file in the pattern instances project. The example shows the remaining tasks, based on the supplied values in the pattern generator.

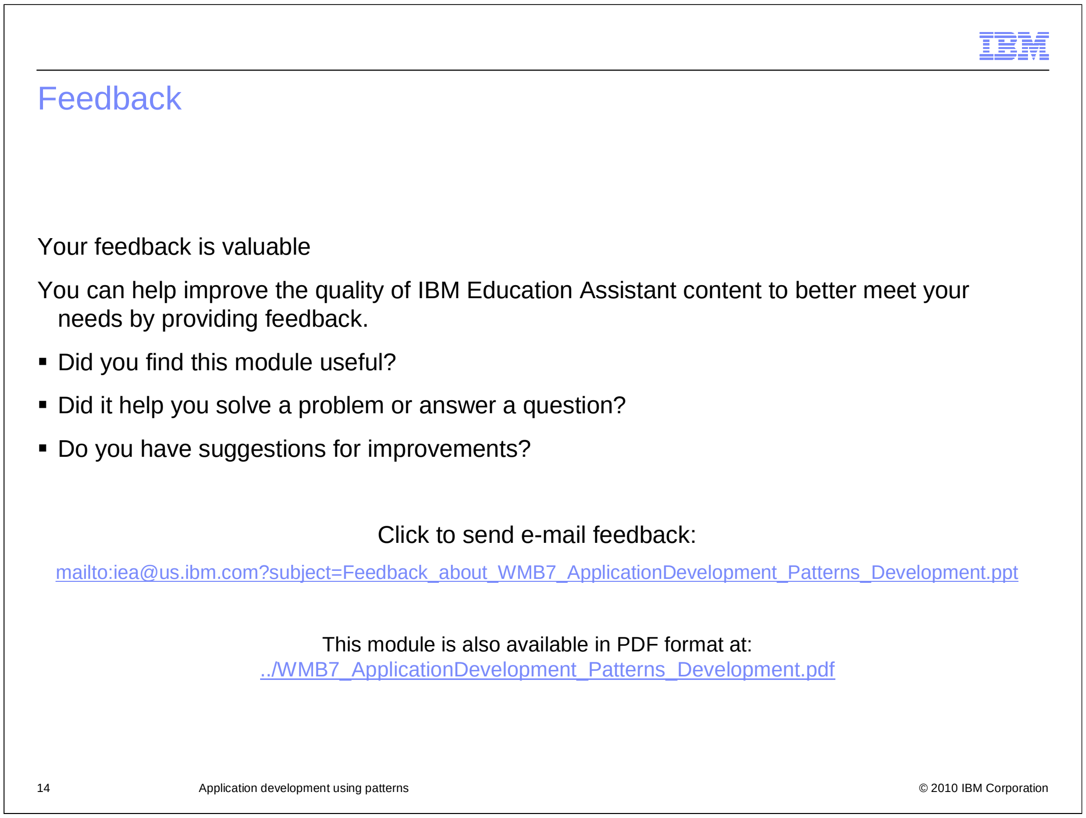


Summary

- Introduction to patterns in Message Broker version 7
- Key concepts
- Patterns in action
- Patterns user interface
  - Navigator view, Patterns Explorer
- Generation and deployment

13 Application development using patterns © 2010 IBM Corporation

In summary, Message Broker version 7 has introduced a new way to generate applications, based on a supplied pattern. These patterns are presented through the pattern explorer, and are used to create message broker applications using the patterns generator. They are tested and deployed using the normal Message Broker tools.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

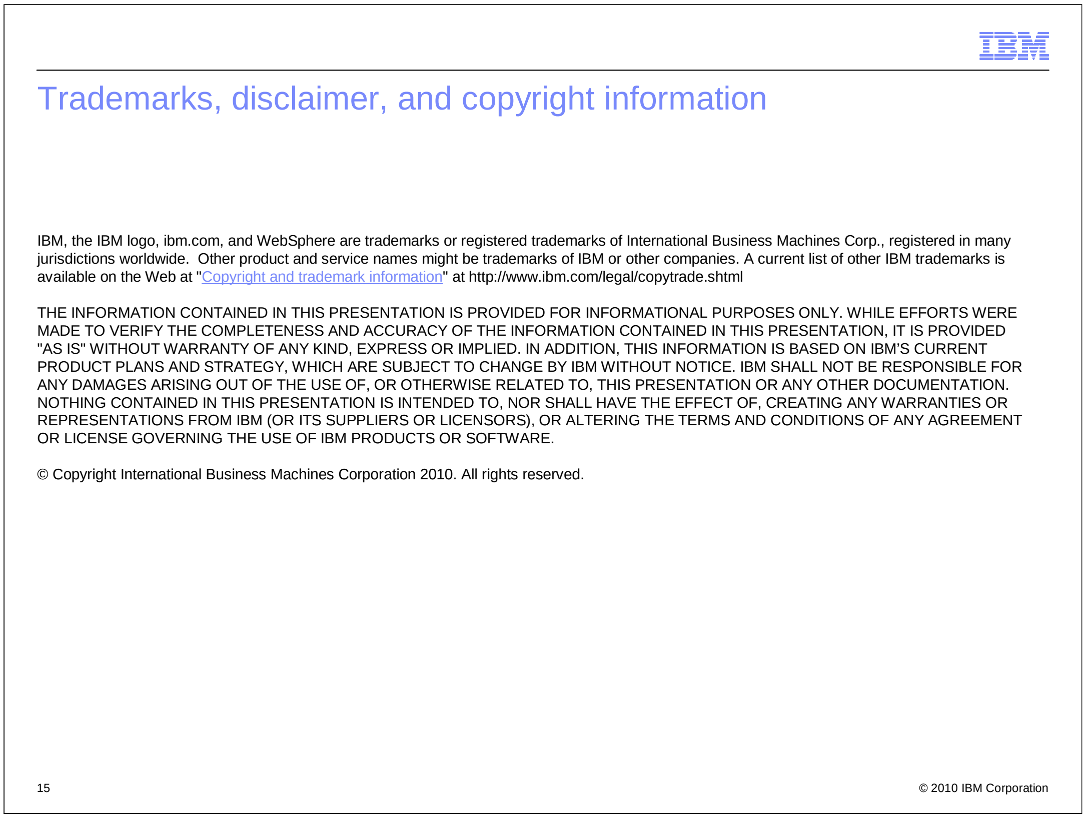
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_ApplicationDevelopment_Patterns_Development.ppt

This module is also available in PDF format at:
../WMB7_ApplicationDevelopment_Patterns_Development.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.