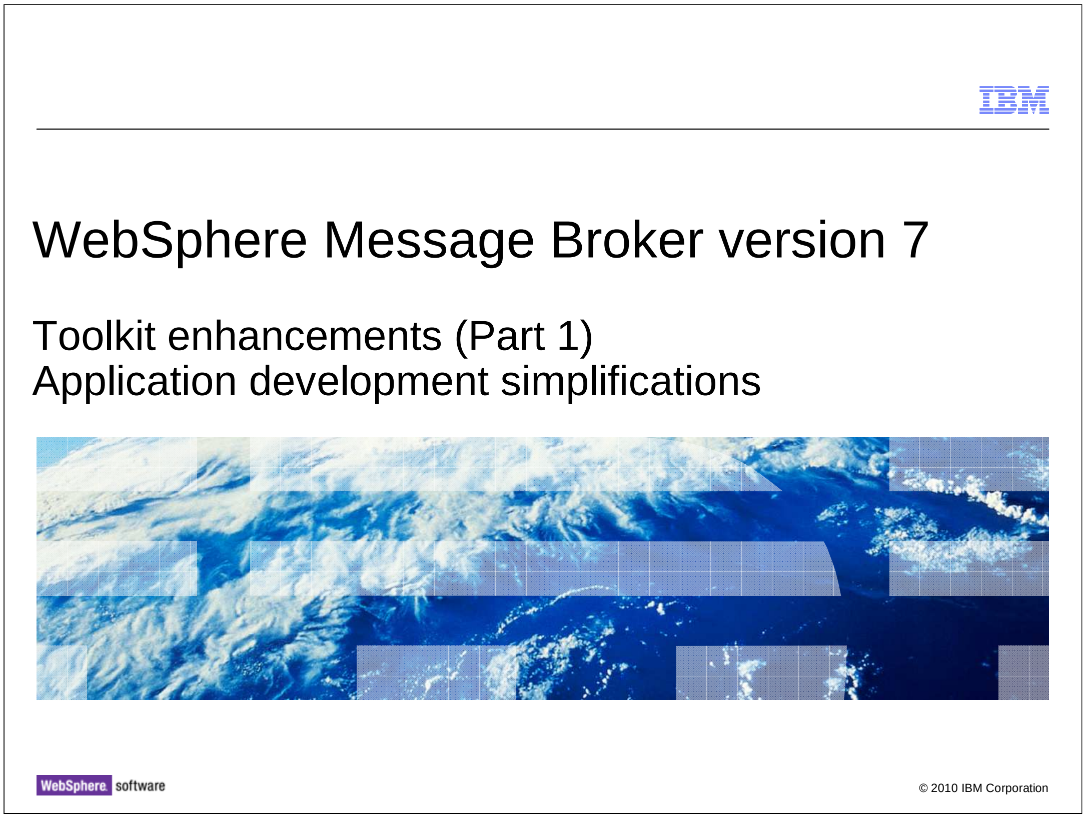
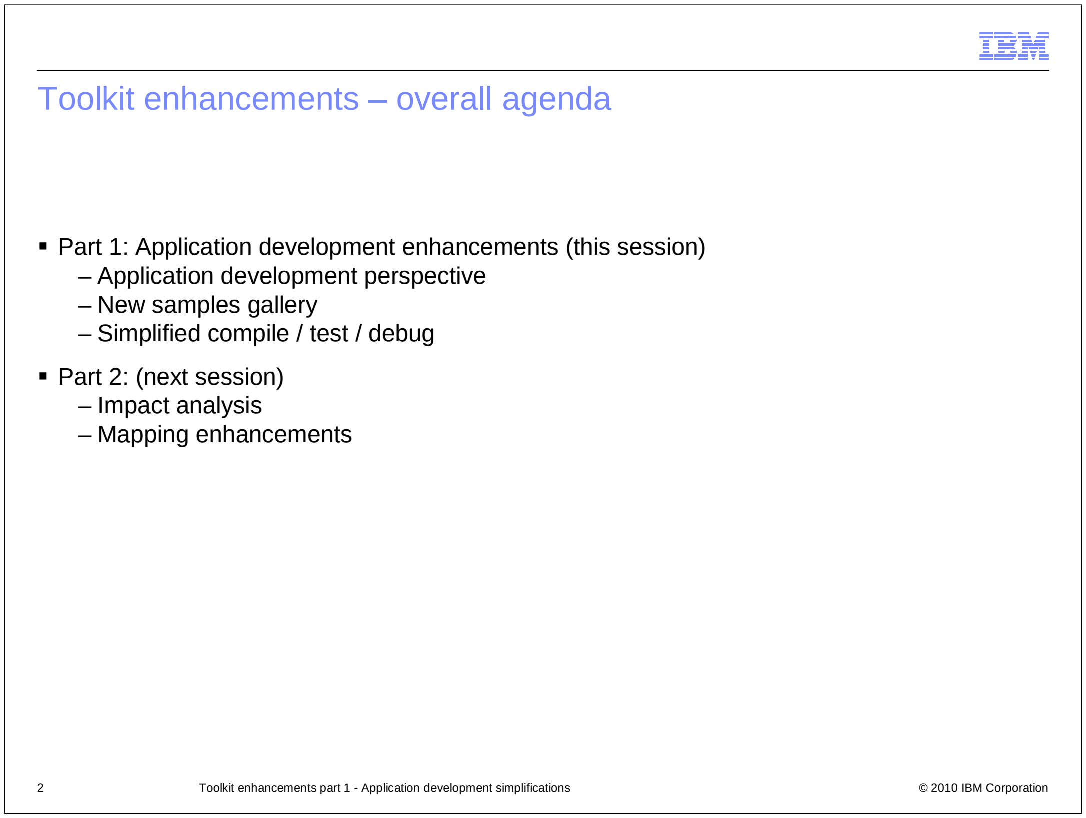


WebSphere Message Broker version 7

Toolkit enhancements (Part 1) Application development simplifications

WebSphere software

This session is the first of two that describe the new toolkit facilities in Message Broker 7.



Toolkit enhancements – overall agenda

- Part 1: Application development enhancements (this session)
  - Application development perspective
  - New samples gallery
  - Simplified compile / test / debug
- Part 2: (next session)
  - Impact analysis
  - Mapping enhancements

This first session will describe the application development perspective, and the changes to the toolkit that have been made in Message Broker version 7. It will also discuss the samples gallery, and the improvements that have been made to the tools for developing, testing and debugging your applications.

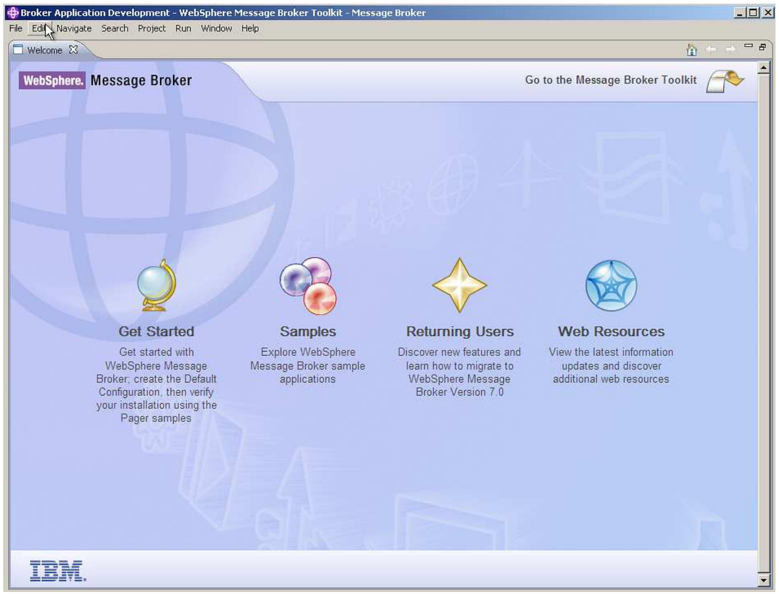
The next session will discuss the additional new functions of impact analysis and enhancements to the mapping node.

As a reminder, and for those new to the Message Broker product, the Message Broker Toolkit is the component that provides the facilities for application construction. It provides the tools to create the artifacts that Message Broker uses. There are several types of application artifact. The two primary types are **Message Sets and Message Flows.**

Message sets contain a business context for a set of messages. They contain all the definitions required to define the logical structure and physical formats for messages used within your applications. They are deployed to the broker runtime component, and are used by the broker to interpret incoming messages and to create outgoing messages.

A message flows is a sequence of processing steps that run in the broker when an input message is received. It’s comprised of several visual components, known as nodes, which are linked with connectors. Message Broker provides many different types of nodes, each of which performs its own function on the incoming message.

Welcome page


Broker Application Development - WebSphere Message Broker Toolkit - Message Broker
File
Edit
Navigate
Search
Project
Run
Window
Help
Welcome
WebSphere. Message Broker
Go to the Message Broker Toolkit
IBM.

Get Started
Get started with WebSphere Message Broker; create the Default Configuration, then verify your installation using the Pager samples

Samples
Explore WebSphere Message Broker sample applications

Returning Users
Discover new features and learn how to migrate to WebSphere Message Broker Version 7.0

Web Resources
View the latest information updates and discover additional web resources

After installing the Message Broker toolkit, and starting the toolkit for the first time, you will see the Welcome page. From here, you have direct links to several topics that can be used to get you started quickly.

First, clicking the “Get Started” icon will take you straight into a dialogue to create the default configuration. This will be particularly useful if you are using a Windows® or Linux® desktop environment, and you plan to perform your development and initial testing on these platforms.

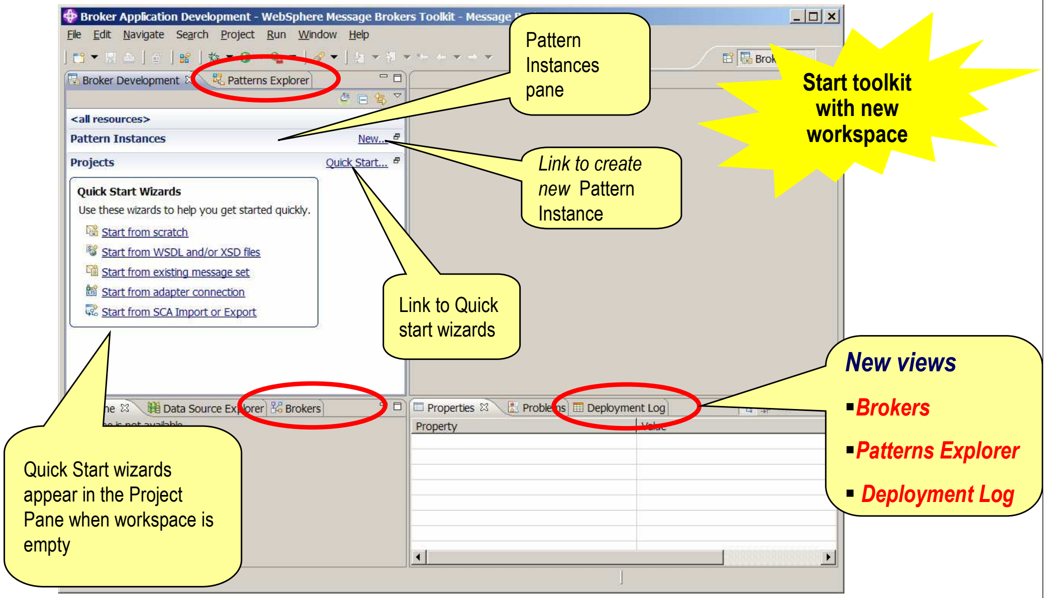
Secondly, you can investigate the supplied sample applications. These are a range of applications which describe how to use most of the application functions provided in Message Broker. All new functions in version 7 have an associated sample, showing how this can be used in an application. Most samples are provided with a script and a pre-built set of test data to show how the sample can be executed.

Next, the “Returning Users” icon is intended for users who already have experience with earlier versions of Message Broker. This icon will directly take you to information about what is new in Message Broker version 7.

Finally, the “Web Resources” icon will present a collection of technical resources on the internet. These include information about Message Broker on the IBM DeveloperWorks Web site, the Message Broker support packs, and how to get additional help with product usage.

If you close the welcome page, by clicking on the small cross on the page tab, you will be taken straight to the Message Broker application development perspective.

Application development perspective – Broker development


Broker Application Development - WebSphere Message Brokers Toolkit - Message B
File Edit Navigate Search Project Run Window Help
Broker Development
Patterns Explorer
<all resources>
Pattern Instances
New...
Projects
Quick Start...
Quick Start Wizards
Use these wizards to help you get started quickly.
Start from scratch
Start from WSDL and/or XSD files
Start from existing message set
Start from adapter connection
Start from SCA Import or Export
Data Source Explorer
Brokers
Properties
Problems
Deployment Log
Property
Value

Pattern Instances pane

Link to create new Pattern Instance

Start toolkit with new workspace

Link to Quick start wizards

Quick Start wizards appear in the Project Pane when workspace is empty

New views

- **Brokers**
- **Patterns Explorer**
- **Deployment Log**

4 Toolkit enhancements part 1 - Application development simplifications

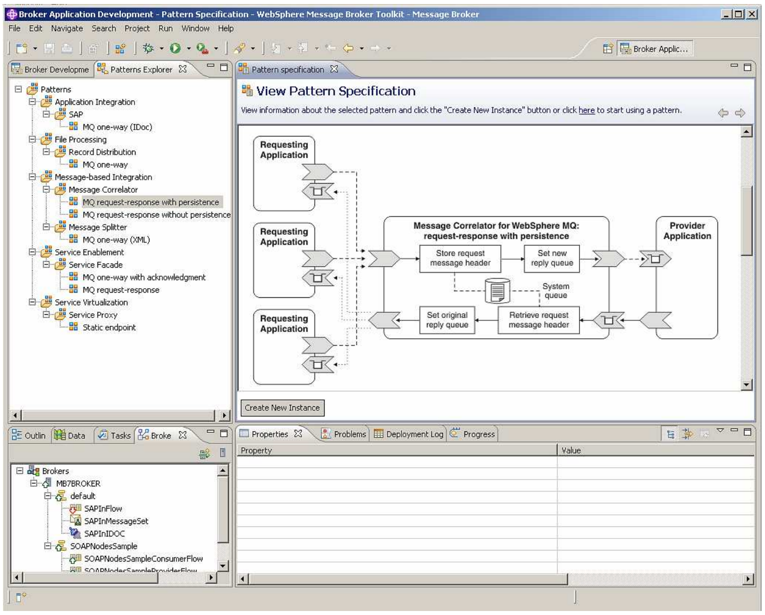
This screen capture shows the application development perspective the first time it has been used after installing Message Broker version 7. When you use version 7 for the first time, you should ensure that you start the toolkit with a new workspace. This will ensure that the Eclipse structure are created correctly.

This screen capture shows the application development primary perspective, which is the primary perspective in Message Broker version 7. If you are familiar with the administration perspective in earlier versions, you will see that this perspective is no longer available. The only other perspective that is used with Message Broker is the debug perspective, described later in this session.

In the top left pane, there are two views. The “Broker Development” view provides the traditional development tools, using the project navigator. All artifacts such as message flows and message sets are located within a project, shown under this navigator. The “Patterns Explorer” view is a new view in version 7, and is used to display the supplied application patterns. Using this view, you can create and generate a new application, based on one of the supplied patterns.

The quick start wizards provide an easy starting point for the creation of new applications, if you are using the traditional style of application development. Depending on the style of application you are using, you can use these wizards to create the required artifacts for your applications. For example, if you are going to connect to SAP using a WebSphere® Adapter, then you can use the “Start from adapter connection” wizard. This will guide you though the development process for the adapter, and resulting message sets and flows.

Application development perspective – Patterns explorer

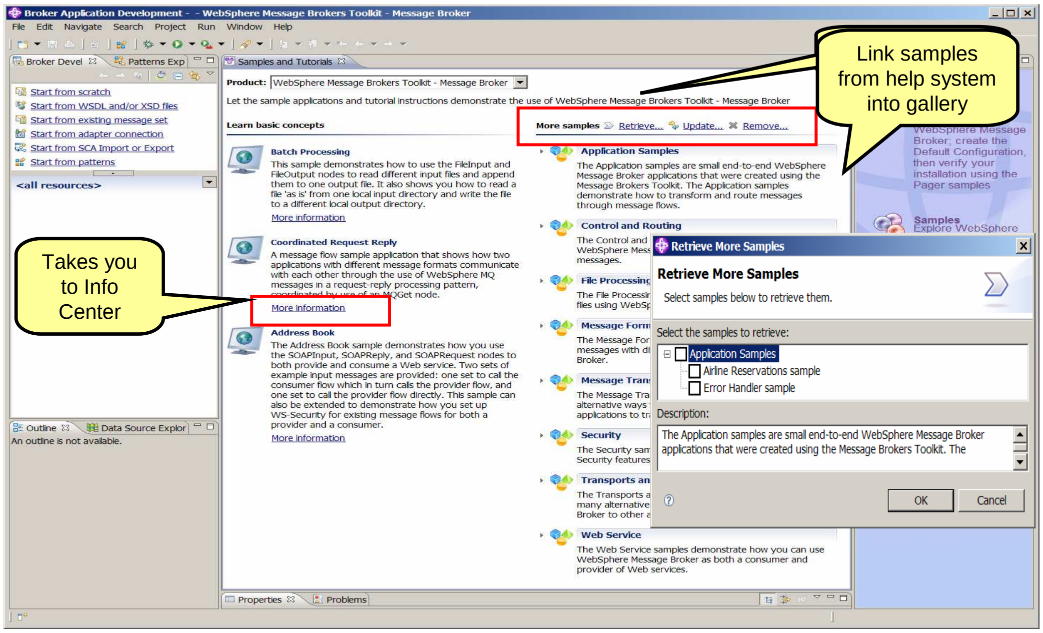


View information about the selected pattern and click the "Create New Instance" button or click here to start using a pattern.

This screen capture shows the “patterns explorer” view. This view is used to browse the supplied patterns, and to generate new message broker applications based on the selected patterns. The example shown is the MQ-based request-response application. The specification view shows a graphic of the pattern. Clicking on the “Create New Instance” button will provide a dialogue to enter properties related to the selected pattern, and will then create a message flow based on this information.

Further information on how to use the patterns-based development is given in a separate IBM Education Assistant session.

Samples gallery



Broker Application Development - - WebSphere Message Brokers Toolkit - Message Broker

An outline is not available.

Product: WebSphere Message Brokers Toolkit - Message Broker

Let the sample applications and tutorial instructions demonstrate the use of WebSphere Message Brokers Toolkit - Message Broker

Learn basic concepts

Batch Processing

This sample demonstrates how to use the FileInput and FileOutput nodes to read different input files and append them to one output file. It also shows you how to read a file 'as is' from one local input directory and write the file to a different local output directory.

More information

Coordinated Request Reply

A message flow sample application that shows how two applications with different message formats communicate with each other through the use of WebSphere MQ messages in a request-reply processing pattern, coordinated by use of an MQGet node.

More information

Address Book

The Address Book sample demonstrates how you use the SOAPInput, SOAPReply, and SOAPRequest nodes to both provide and consume a Web service. Two sets of example input messages are provided: one set to call the consumer flow which in turn calls the provider flow, and one set to call the provider flow directly. This sample can also be extended to demonstrate how you set up WS-Security for existing message flows for both a provider and a consumer.

More information

More samples Retrieve... Update... Remove...

Application Samples

The Application samples are small end-to-end WebSphere Message Broker applications that were created using the Message Brokers Toolkit. The Application samples demonstrate how to transform and route messages through message flows.

Control and Routing

The Control and WebSphere Mess messages.

File Processing

The File Processir files using WebSp

Message Form

The Message For messages with di Broker.

Message Trans

The Message Tra alternative ways applications to tr

Security

The Security sam Security features

Transports an

The Transports a many alternative Broker to other a

Web Service

The Web Service samples demonstrate how you can use WebSphere Message Broker as both a consumer and provider of Web services.

WebSphere Message Broker; create the Default Configuration, then verify your installation using the Pager samples

Samples Explore WebSphere

Retrieve More Samples

Retrieve More Samples

Select samples below to retrieve them.

Select the samples to retrieve:

- □ Application Samples
  - □ Airline Reservations sample
  - □ Error Handler sample

Description:

The Application samples are small end-to-end WebSphere Message Broker applications that were created using the Message Brokers Toolkit. The

Link samples from help system into gallery

Takes you to Info Center

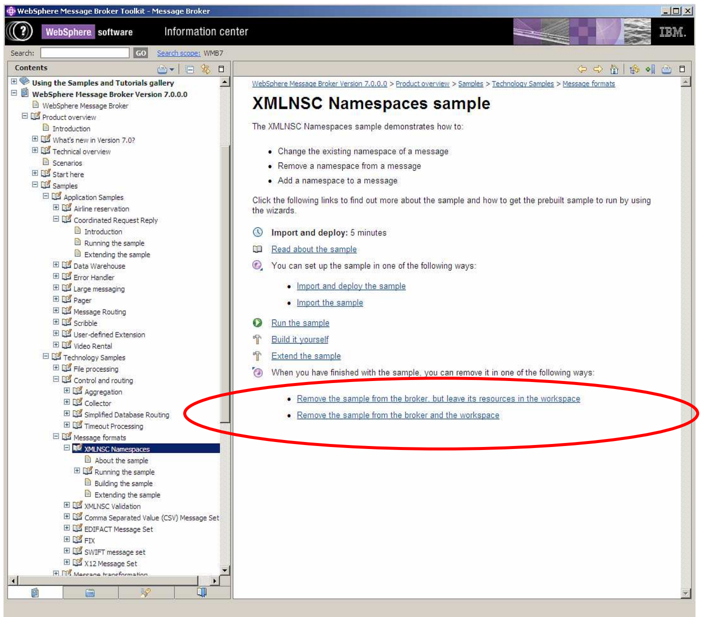
6 Toolkit enhancements part 1 - Application development simplifications © 2010 IBM Corporation

The samples gallery contains many applications that show you how to use many of the applications function in Message Broker. It is accessed by clicking “Help”, then “Samples and Tutorials”, then “Message Broker Toolkit, Message Broker”. The main samples menu are shown as on this screen capture.

This view has a consistent appearance for the WebSphere family of products. If you have installed the shell share environment, you can also view the samples for all the WebSphere Process Management products.

To obtain further information about each of the samples, and to import the sample into the toolkit, click “More information” for the required sample. This will open the information center at the place where the samples are described in detail, as shown on the next slide.

Samples gallery – Import and deploy samples



WebSphere Message Broker Version 7.0.0.0 > Product overview > Samples > Technology Samples > Message formats

XMLNSC Namespaces sample

The XMLNSC Namespaces sample demonstrates how to:

- Change the existing namespace of a message
- Remove a namespace from a message
- Add a namespace to a message

Click the following links to find out more about the sample and how to get the prebuilt sample to run by using the wizards.

Import and deploy: 5 minutes

Read about the sample

You can set up the sample in one of the following ways:

- Import and deploy the sample
- Import the sample

Run the sample

Build it yourself

Extend the sample

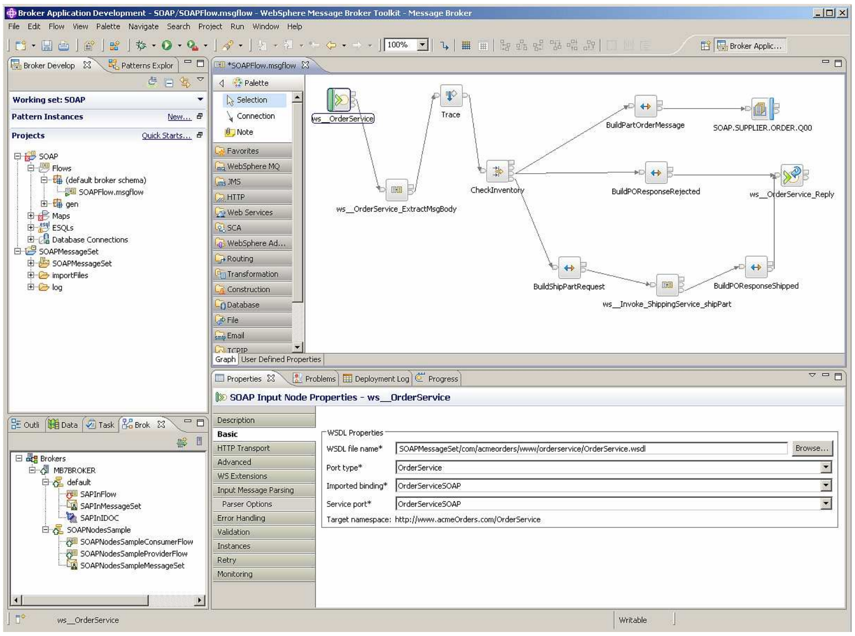
When you have finished with the sample, you can remove it in one of the following ways:

- Remove the sample from the broker, but leave its resources in the workspace
- Remove the sample from the broker and the workspace

When you select a particular sample, the Message Broker information center is opened, and all available samples will be shown in the menu in the navigator. The selected sample will be shown on the right, and you can import the sample into the toolkit workspace from here. If required, you can also deploy the sample directly into the broker runtime. This action will also create the required runtime artifacts, such as MQ queues. To do this, you must have created the default configuration, and have a broker named MB7BROKER.

New in version 7 is the option to remove the selected sample. You can either remove the deployed sample from the broker, or remove it from the toolkit workspace as well.

The application development perspective (compile/test/debug)



SOAP Input Node Properties - ws__OrderService

WSDL Properties

WSDL file name* SOAPMessageSet/com/acmeorders/www/orderservice/OrderService.wsdl

Port type* OrderService

Imported binding* OrderServiceSOAP

Service port* OrderServiceSOAP

Target namespace: http://www.acmeOrders.com/OrderService

ws__OrderService

This screen capture shows the application development perspective, and shows the most common set of views that you are likely to use. In the top left is the Broker Development view, showing the project navigator for all projects in the workspace. In the example shown, a working set called SOAP has been defined, which shows just the projects that are contained within this working set. Working sets is a simple technique to reduce the number of artifacts that are displayed in the project navigator. If you want to see all projects, select the option to display “all resources” from the working set pull down.

The top right shows the selected message flow artifact. In this example, the SOAPFlow message flow is selected in the navigator and displayed in the flow editor.

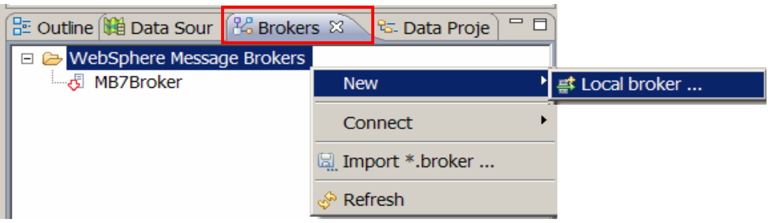
The bottom right has a several available views. In the example, the properties of the message flow input node are shown. In this case, this is a SOAP input node, so the various property tabs relate to this type of input.

In this area, you can also select the Problems tab if it is available. The Deployment Log tab is used when deploying applications to the broker.

In the bottom left, there are again several available views. The most useful for the message flow developer is the Brokers view. This replaces the administration perspective in earlier versions of Message Broker. This is described in more detail on the next slide.

Brokers view

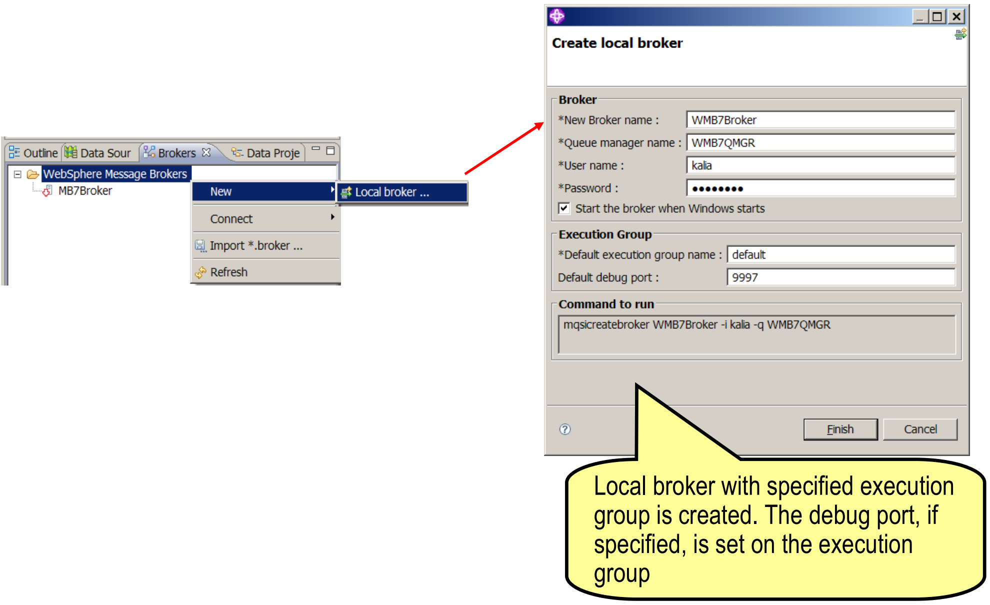
- *Brokers view* hosted along side O*utline view* allows application developer to do following tasks
  - Create / delete local brokers
  - Connect to remote broker
  - Create / delete execution group
  - Configure debug port and launch debugger for execution group
  - Deploy message flows to execution group
  - Remove deployed artifacts from execution group
  - Start / stop brokers, execution groups, and message flows



The Brokers view is a new view in Message Broker version 7, and is provided to enable an application developer to define the resources needed to run an application. It is also used to deploy a new or changed application into the required broker.

Using this view, you can perform the majority of the functions previously available in the administration perspective. You can also specify information required for application debugging, such as setting the debug port numbers.

Create new local broker



Create local broker

Broker

*New Broker name : WMB7Broker

*Queue manager name : WMB7QMGR

*User name : kalia

*Password : ••••••••

☑ Start the broker when Windows starts

Execution Group

*Default execution group name : default

Default debug port : 9997

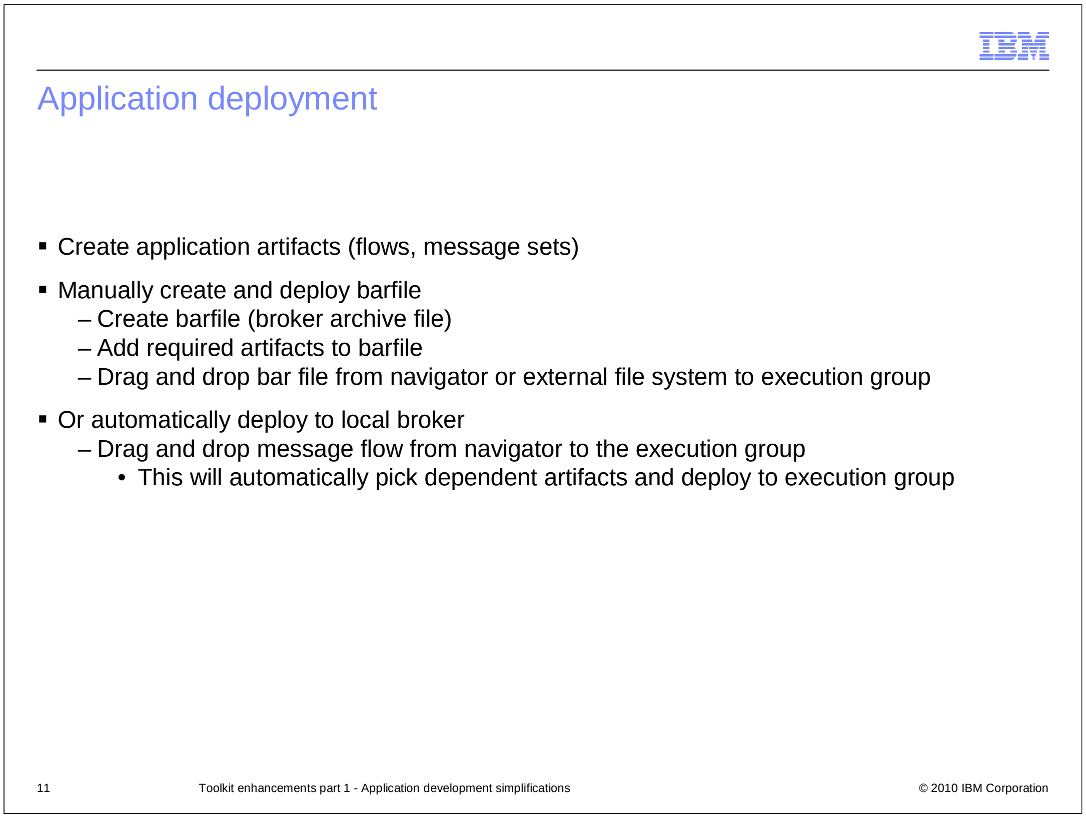
Command to run

```
mqsicreatebroker WMB7Broker -i kalia -q WMB7QMGR
```

Local broker with specified execution group is created. The debug port, if specified, is set on the execution group

If you selected the option to create a new local broker, the window shown here will be displayed. Specify the name and connection details for the new broker. If required, specify the default port for message flow debugging. Only one execution group will be created initially. Additional execution groups can be created later.

It is not possible to create a remote broker using this tool.



Application deployment

- Create application artifacts (flows, message sets)
- Manually create and deploy barfile
  - – Create barfile (broker archive file)
  - – Add required artifacts to barfile
  - – Drag and drop bar file from navigator or external file system to execution group
- Or automatically deploy to local broker
  - – Drag and drop message flow from navigator to the execution group
    - This will automatically pick dependent artifacts and deploy to execution group

11 Toolkit enhancements part 1 - Application development simplifications © 2010 IBM Corporation

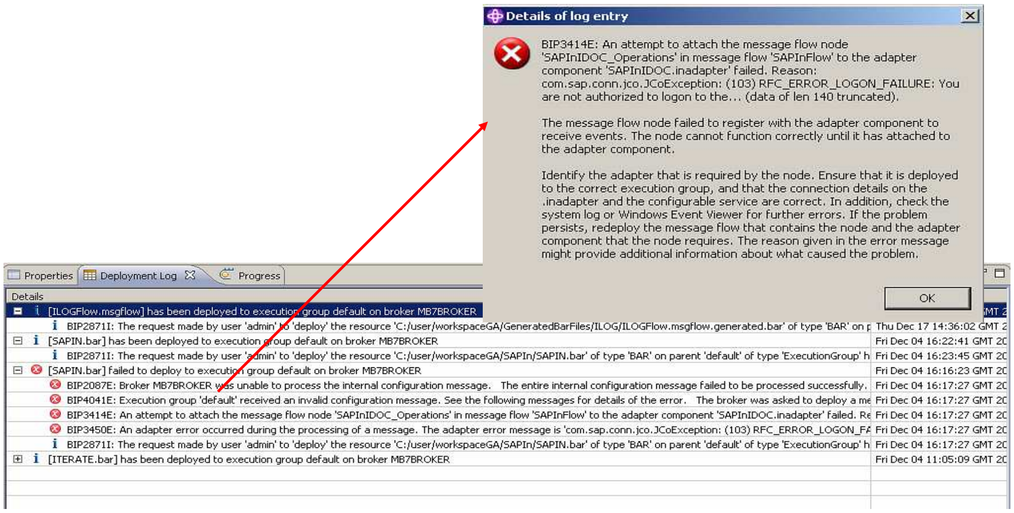
After creating the various artifacts required for a Message Broker application, such as message flows, message sets and adapter definitions, these must be deployed to the broker runtime. All the required artifacts are added to a broker archive file, normally known as a barfile. This can be created manually in the project navigator. When it has been created, each project artifact is added to the barfile, which is then saved.

To deploy the application, you can drop the barfile onto the intended execution group. You can either use the barfile that you have just created in the project navigator, or you can deploy a barfile that you have previously created and stored on an external file system.

You can also deploy a barfile by right-clicking the barfile in the project navigator, and selecting deploy, then selecting the target execution group. Alternatively, you can start with the execution group, right-clicking, selecting deploy, and selecting the required barfile. In the second case, you can select barfiles from outside the toolkit workspace.

In the early stages of development of an application, you might find it quicker to deploy the message flow directly onto the execution group, without creating the barfile. In this case, the toolkit will automatically generate a barfile, and include all required artifacts, such as dependent message sets, to enable the application to be deployed. The generated barfile will be stored in the message flow project, under the directory “GeneratedBarFiles”.

Application deployment log and detailed messages



BIP3414E: An attempt to attach the message flow node 'SAPInIDOC_Operations' in message flow 'SAPInFlow' to the adapter component 'SAPInIDOC.inadapter' failed. Reason: com.sap.conn.jco.JCoException: (103) RFC_ERROR_LOGON_FAILURE: You are not authorized to logon to the... (data of len 140 truncated).

The message flow node failed to register with the adapter component to receive events. The node cannot function correctly until it has attached to the adapter component.

Identify the adapter that is required by the node. Ensure that it is deployed to the correct execution group, and that the connection details on the .inadapter and the configurable service are correct. In addition, check the system log or Windows Event Viewer for further errors. If the problem persists, redeploy the message flow that contains the node and the adapter component that the node requires. The reason given in the error message might provide additional information about what caused the problem.

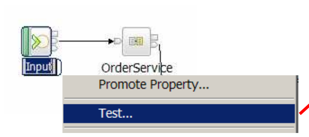
```
[ILOGFlow.msgflow] has been deployed to execution group default on broker MB7BROKER                                                  GMT 2
    BIP2871I: The request made by user 'admin' to 'deploy' the resource 'C:/user/workspaceGA/GeneratedBarFiles/ILOG/ILOGFlow.msgflow.generated.bar' of type 'BAR' on p   Thu Dec 17 14:36:02 GMT 2
[SAPIN.bar] has been deployed to execution group default on broker MB7BROKER                                                         Fri Dec 04 16:22:41 GMT 20
    BIP2871I: The request made by user 'admin' to 'deploy' the resource 'C:/user/workspaceGA/SAPIn/SAPIN.bar' of type 'BAR' on parent 'default' of type 'ExecutionGroup' h   Fri Dec 04 16:23:45 GMT 20
[SAPIN.bar] failed to deploy to execution group default on broker MB7BROKER                                                          Fri Dec 04 16:16:23 GMT 20
    BIP2087E: Broker MB7BROKER was unable to process the internal configuration message.   The entire internal configuration message failed to be processed successfully.   Fri Dec 04 16:17:27 GMT 20
    BIP4041E: Execution group 'default' received an invalid configuration message. See the following messages for details of the error.   The broker was asked to deploy a me   Fri Dec 04 16:17:27 GMT 20
    BIP3414E: An attempt to attach the message flow node 'SAPInIDOC_Operations' in message flow 'SAPInFlow' to the adapter component 'SAPInIDOC.inadapter' failed. Re   Fri Dec 04 16:17:27 GMT 20
    BIP3450E: An adapter error occurred during the processing of a message. The adapter error message is 'com.sap.conn.jco.JCoException: (103) RFC_ERROR_LOGON_FA   Fri Dec 04 16:17:27 GMT 20
    BIP2871I: The request made by user 'admin' to 'deploy' the resource 'C:/user/workspaceGA/SAPIn/SAPIN.bar' of type 'BAR' on parent 'default' of type 'ExecutionGroup' h   Fri Dec 04 16:17:27 GMT 20
[ITERATE.bar] has been deployed to execution group default on broker MB7BROKER                                                       Fri Dec 04 11:05:09 GMT 20
```

Message Broker version 7 has changed the way in which barfiles are deployed to the broker. Previous versions used an asynchronous deployment tool, which used the configuration manager, an intermediate component between the Toolkit and the broker runtime. Version 7 now enables the toolkit to connect directly to the target broker, so deployment operations are done synchronously. When you deploy a barfile from the toolkit, the toolkit will wait for the deployment operation to complete. When it is complete, the results of the deployment are shown in the deployment log, which is one of the views in the application development perspective. This will normally be found in the bottom right area.

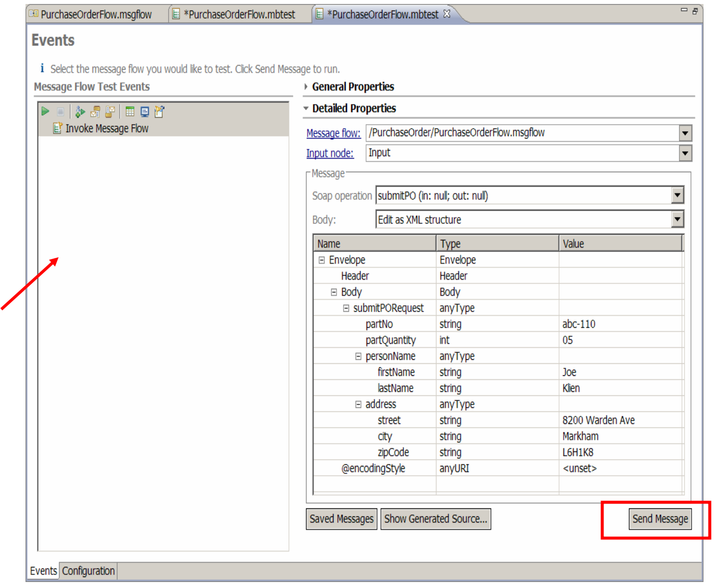
The information presented here now includes information on who performed the deployment operation, the name of the deployed barfile, and details of any deployment failure if appropriate. The example shown on this screen capture shows a failure to deploy the SAPIn barfile. Double-clicking each of the messages in the deployment log will show further details of the deployment. In this example, the SAPIn barfile failed to deploy because the security credentials of the required adapter for SAP were incorrect.

Test / debug using test client (1 of 3)

Launch test client from the message flow editor or MB navigator



- Compose test message in test client
- Click “Send” button


PurchaseOrderFlow.msgflow
*PurchaseOrderFlow.mbtest
*PurchaseOrderFlow.mbtest

Events

Select the message flow you would like to test. Click Send Message to run.

Message Flow Test Events

Invoke Message Flow

General Properties

Detailed Properties

Message flow: /PurchaseOrder/PurchaseOrderFlow.msgflow

Input node: Input

Message

Soap operation submitPO (in: null; out: null)

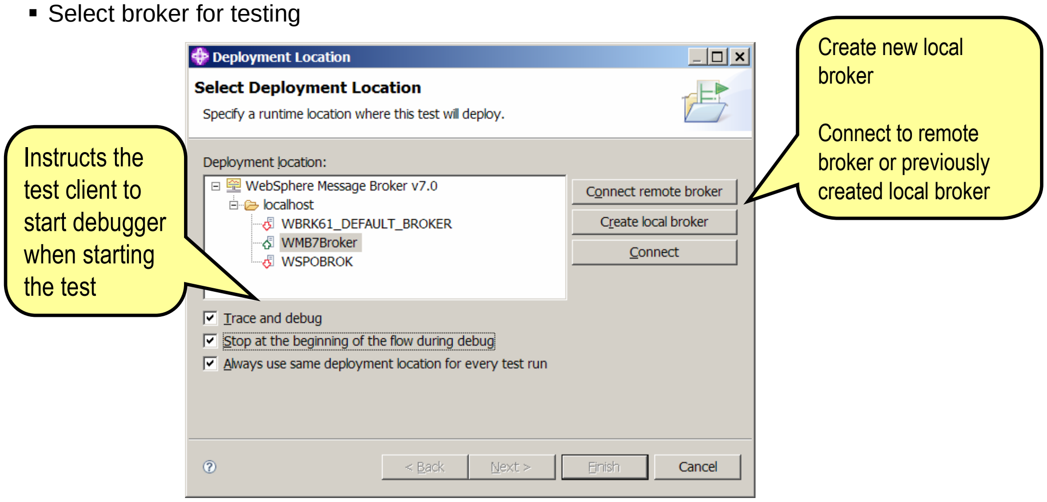
Body: Edit as XML structure

Name	Type	Value
Envelope	Envelope	
Header	Header	
Body	Body	
submitPORequest	anyType	
partNo	string	abc-110
partQuantity	int	05
personName	anyType	
firstName	string	Joe
lastName	string	Klien
address	anyType	
street	string	8200 Warden Ave
city	string	Markham
zipCode	string	L6H1K8
@encodingStyle	anyURI	<unset>

Message Broker application can be tested with the built-in Test Client. The Test Client is launched by right-clicking on the first node of the required message flow, and selecting Test from the pop-up menu. In the Test Client window, you can specify the execution parameters, including the required input message. When the input data has been created, click the “Send Message” button to initiate the test.

Test / debug using test client (2 of 3)

- Select broker for testing



Deployment Location

Select Deployment Location

Specify a runtime location where this test will deploy.

Deployment location:

```
⊟ WebSphere Message Broker v7.0
  ⊟ localhost
      WBRK61_DEFAULT_BROKER
      WMB7Broker
      WSPOBROK
```

☑ Trace and debug

☑ Stop at the beginning of the flow during debug

☑ Always use same deployment location for every test run

Instructs the test client to start debugger when starting the test

Create new local broker

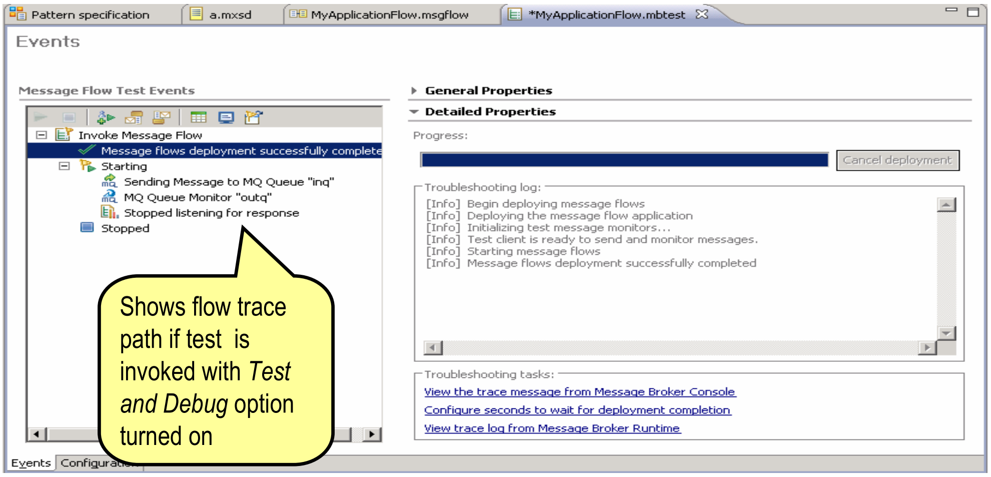
Connect to remote broker or previously created local broker

The Test Client will then ask you which broker and which execution group you are going to use to perform the test. If the broker does not exist at this time, you can create it now. Alternatively, if the broker is a remote broker, you can connect to it at this time. Then select the target execution group, and click Finish. If the connected broker has more than one execution group, the first execution group will be selected. You can change this selection as required.

When using the Test Client, you can also run the debugger at the same time. Check the appropriate tick boxes to activate the debug function.

Test / debug using test client (3 of 3)

- Test client will deploy message flow, send test message and receive reply
  - If *test and debug* option is selected, then debugger will be launched when test starts. Flow execution will stop at user defined breakpoints



```
⊟ Invoke Message Flow
    Message flows deployment successfully complete
  ⊟ Starting
      Sending Message to MQ Queue "inq"
      MQ Queue Monitor "outq"
      Stopped listening for response
    Stopped
```

Troubleshooting log:

```
[Info]  Begin deploying message flows
[Info]  Deploying the message flow application
[Info]  Initializing test message monitors...
[Info]  Test client is ready to send and monitor messages.
[Info]  Starting message flows
[Info]  Message flows deployment successfully completed
```

Troubleshooting tasks:

View the trace message from Message Broker Console
Configure seconds to wait for deployment completion
View trace log from Message Broker Runtime

Shows flow trace path if test is invoked with *Test and Debug* option turned on

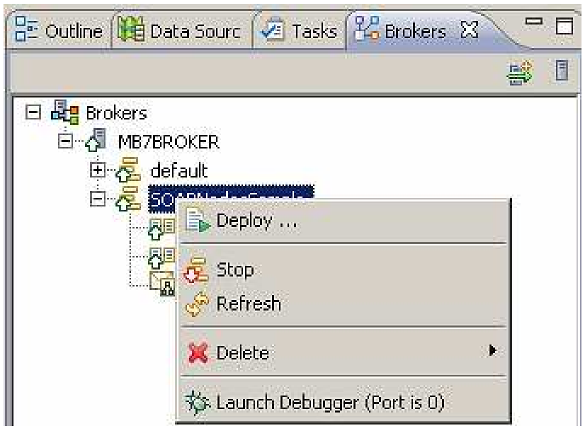
When you have made the broker and execution group selections, clicking finish will deploy the application, and start execution of the message flow. You will see a window similar to the screen capture shown here. If you have activated the debug function, and you have defined breakpoints in your application, then execution will stop at the defined breakpoints. The toolkit perspective will switch to debug, and you will see the debug tools such as the in-flight message tree data, and line-by-line control of code execution. This screen capture shows the events tab, which is where you can control execution of the message flow.

If you have selected the flow trace option, imbedded trace information is presented in the message flow events panel on the left side.

The Test Client will monitor all MQ output queues in the message flow. The default action is to stop monitoring when the first MQ output message is received. This action can be over-ridden in the configuration view, using the tab in the bottom left corner of the Test Client window.

Launching debugger

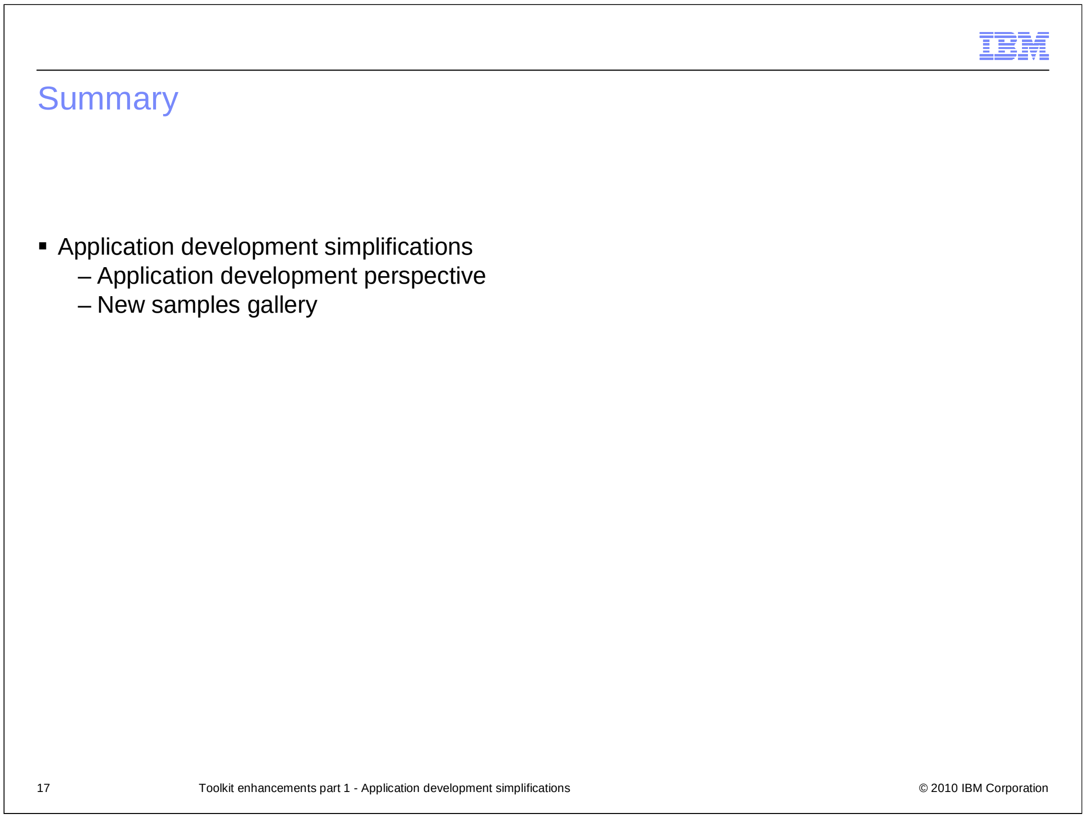
- Configure debug port on execution group
  - – No need to issue command from broker command console to set debug port
- Launch debugger - No need to create debug launch configuration



You can use the message flow debugger independently of the Test Client. Message flows can be invoked in a variety of ways. For example, an MQ application can be used to place messages on the input queue of the MQ input node, or an application generating HTTP requests can directly place messages on an HTTP input node.

To start the debugger in the Message Broker toolkit, right-click the required execution group, and select Launch Debugger from the pop-up menu. If you have not previously specified the debugger port, then this will be set to 0, as shown in this screen capture. To activate the debugger, specify a debug port, and click OK in the resulting window. It is no longer necessary to perform this action from the command line, as in earlier versions.

In version 7, there is also no need to create a debug launch configuration.



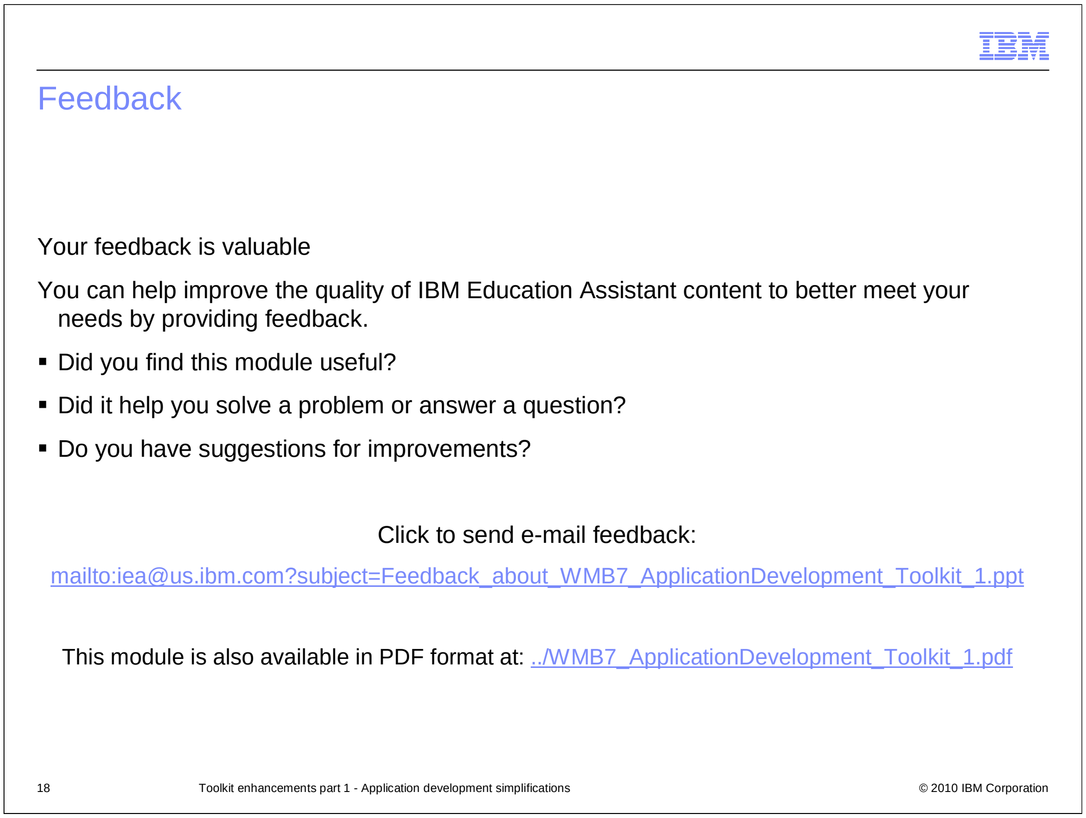
Summary

- Application development simplifications
  - Application development perspective
  - New samples gallery

17 Toolkit enhancements part 1 - Application development simplifications

In summary, Message Broker version 7 has introduced significant enhancements in the application development toolkit. There is now just one primary toolkit perspective, and all application development functions can be done from this perspective.

The samples gallery has been extended, and all new product functions have a corresponding sample in the gallery.



Feedback

Your feedback is valuable

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

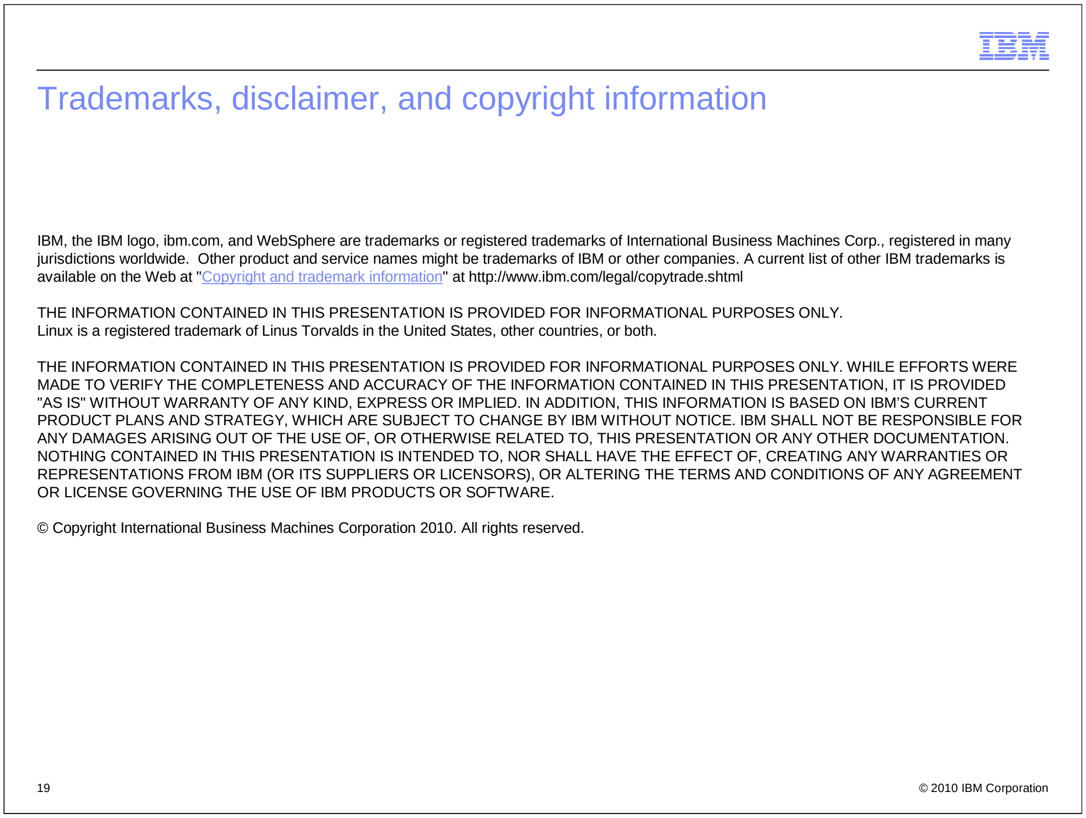
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_ApplicationDevelopment_Toolkit_1.ppt

This module is also available in PDF format at: ../WMB7_ApplicationDevelopment_Toolkit_1.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.

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.