

WebSphere Message Broker



# Introduction

*Version 6 Release 0*



WebSphere Message Broker



# Introduction

*Version 6 Release 0*

**Note**

Before using this information and the product it supports, read the information in the Notices appendix.

This edition applies to version 6, release 0, modification 0, fix pack 4 of IBM® WebSphere® Message Broker, with Message Brokers Toolkit version 6, release 0, modification 2, and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2000, 2007. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

**About this topic collection. . . . . v**

---

**Part 1. Start here . . . . . 1**

<b>Welcome to WebSphere Message Broker</b>	
<b>Version 6.0 . . . . . 3</b>	
How do I use the information center? . . . . . 4	
Where can I get an overview of WebSphere Message Broker? . . . . . 6	
How do I use the Message Brokers Toolkit? . . . . . 7	
How can I check that my installation was successful? . . . . . 8	
How do I design and develop applications? . . . . . 9	
WebSphere Message Broker Administration. . . . . 24	
How can I diagnose problems? . . . . . 31	
What information is available for users of previous versions? . . . . . 33	

---

**Part 2. Product overview . . . . . 35**

<b>Product overview . . . . . 37</b>	
WebSphere Message Brokers introduction . . . . . 37	

WebSphere Message Broker Version 6.0 . . . . . 57	
WebSphere Message Broker scenario . . . . . 74	
Accessibility features for WebSphere Message Broker . . . . . 76	

---

**Part 3. Reference . . . . . 79**

<b>Workbench . . . . . 81</b>	
Perspectives in the Message Brokers Toolkit . . . . . 81	
Editors in the Message Brokers Toolkit . . . . . 90	
Resource types in the Message Brokers Toolkit . . . . . 112	
Message Brokers Toolkit keyboard shortcuts . . . . . 115	

**Glossary of terms and abbreviations 119**

---

**Part 4. Appendixes . . . . . 137**

<b>Appendix. Notices . . . . . 139</b>	
Trademarks . . . . . 141	

**Index . . . . . 143**



---

## About this topic collection

This PDF has been created from the WebSphere Message Broker Version 6.0 (Fix Pack 4 update, March 2007) information center topics. Always refer to the WebSphere Message Broker online information center to access the most current information. The information center is periodically updated on the document update site and this PDF and others that you can download from that Web site might not contain the most current information.

The topic content included in the PDF does not include the "Related Links" sections provided in the online topics. Links within the topic content itself are included, but are active only if they link to another topic in the same PDF collection. Links to topics outside this topic collection are also shown, but these attempt to link to a PDF that is called after the topic identifier (for example, ac12340\_.pdf) and therefore fail. Use the online information to navigate freely between topics.

**Feedback:** do not provide feedback on this PDF. Refer to the online information to ensure that you have access to the most current information, and use the Feedback link that appears at the end of each topic to report any errors or suggestions for improvement. Using the Feedback link provides precise information about the location of your comment.

The content of these topics is created for viewing online; you might find that the formatting and presentation of some figures, tables, examples, and so on are not optimized for the printed page. Text highlighting might also have a different appearance.





---

## Part 1. Start here

<b>Welcome to WebSphere Message Broker Version</b>		How do I use ESQL? . . . . .	15
<b>6.0</b> . . . . .	3	How do I use Java?. . . . .	16
How do I use the information center? . . . . .	4	How do I use XML transformations? . . . . .	17
Where can I find other sources of help		How do I design a message model? . . . . .	18
information? . . . . .	5	How do I develop a message model?. . . . .	19
Where can I get an overview of WebSphere Message		How do I develop publish/subscribe	
Broker? . . . . .	6	applications? . . . . .	21
How do I use the Message Brokers Toolkit? . . . . .	7	How do I deploy and test applications? . . . . .	22
How can I check that my installation was successful? . . . . .	8	WebSphere Message Broker Administration. . . . .	24
How do I design and develop applications? . . . . .	9	How do I plan and configure a broker domain? . . . . .	24
What do I need to know to start developing		How do I deploy and configure applications? . . . . .	27
applications? . . . . .	9	How do I administer and monitor a broker	
Basic application development concepts . . . . .	9	domain? . . . . .	30
Advanced application development concepts . . . . .	11	How can I diagnose problems?. . . . .	31
How do I construct message flows? . . . . .	12	What information is available for users of previous	
How do I program message flows? . . . . .	14	versions? . . . . .	33
How do I use message mappings?. . . . .	14		








---

# Welcome to WebSphere Message Broker Version 6.0

New users: use this topic to help you navigate through the information center:

- If you are new to WebSphere Message Broker, take a look at the Getting started section first.
- If you are ready to start developing message broker applications, take a look at the Developing applications section.
- If you are ready to design and configure your broker domain, take a look at the WebSphere Message Broker administration section.
- The Additional information section includes links that might be of interest to all users.

Use the links in the following sections to go to start here topics that contain useful links into the information center or to external information. You can recognize the start here topics by the blue background, as shown in this paragraph. To return to this page, click **Navigation view** () to display the contents of the information center, and then select Start here. To find out where the page that you are currently viewing is located in the contents of the information center, click **Show in Table of Contents** (). You can use **Back** () to go back through the pages you have previously viewed.

Some of the links in the start here topics work only if you are accessing the WebSphere Message Broker documentation from the Message Brokers Toolkit. The links that work only in the Message Brokers Toolkit are displayed with this icon . Some of the links in the start here topics work only if you have a connection to the internet, and these links are displayed using this icon .

## Getting started

WebSphere Message Broker Version 6.0 provides an advanced enterprise service bus, delivering universal connectivity and data transformation. The following introductory information describes the high level concepts of the product and some basic tasks, including how to use this information center and how to use the Message Brokers Toolkit.

- “How do I use the information center?” on page 4 Find out how to use this information center, including navigation and searching techniques.
- “Where can I get an overview of WebSphere Message Broker?” on page 6 Get links to introductory information about WebSphere Message Broker.
- “How do I use the Message Brokers Toolkit?” on page 7 Find out how to use the Message Brokers Toolkit.
- “How can I check that my installation was successful?” on page 8 Find out how to check your installation was successful.

## Developing Applications

Use the links in this section to help you start developing your own WebSphere Message Broker applications. Links are also included to help you understand message broker concepts that might be useful when you develop your own

applications. You can also use the WebSphere Message Broker Samples to help you develop your own applications. Most of the samples provide working examples of applications and demonstrate the features available in WebSphere Message Broker. In order to run samples or deploy and test your own applications, you must have created a broker domain.

You can use the Default Configuration wizard to create a basic broker domain for testing applications.

- “What do I need to know to start developing applications?” on page 9 Find out about the WebSphere Message Broker concepts you might want to know before developing your own applications.
- “How do I design and develop applications?” on page 9 Get started developing your own message flow applications and associated resources.
- “How do I deploy and test applications?” on page 22 Find out how to deploy your applications to a broker and how to test them.
- How do I debug and troubleshoot my applications? Find out how to debug message flow applications and troubleshoot problems.

## WebSphere Message Broker Administration

In order to test and run your message broker applications you must set up a broker domain. The broker domain is one or more brokers that share a common configuration, together with a Configuration Manager that controls them. Use the links in this section to find out how to plan, configure and administer your broker domain.

- “How do I plan and configure a broker domain?” on page 24 Find out how to plan and configure a broker domain for your business.
- “How do I deploy and configure applications?” on page 27 Find out how to configure your applications and get them running.
- “How do I administer and monitor a broker domain?” on page 30 Find out about day to day administrative tasks and about monitoring the performance of your applications.

## Additional Information

Use the following links to get additional information about using WebSphere Message Broker:

- Where can I find out about the accessibility functions of WebSphere Message Broker? Find out about accessibility features in WebSphere Message Broker and shortcut keys to use in the information center.
- How can I troubleshoot any problems? Find out how to debug message flow applications and troubleshoot problems.
- “What information is available for users of previous versions?” on page 33 Find out about migration and what’s new for users of previous versions of the product.

---

## How do I use the information center?

New users: Use this topic to find out how to use the WebSphere Message Broker information center.

- What is the information center?  
Find out what the information center is, and the alternative ways that you can use to access it.

- How do I navigate the information center?  
Learn how to find your way around the information center.
- What are the accessibility features and keyboard shortcuts in the information center?  
Find out about accessibility in the information center and how to navigate using keyboard shortcuts.
- How do I search the information center?  
Learn how to search the information center, limiting your search scope and making your searches more effective.
- How do I set bookmarks and print topics?  
Find out how to bookmark information to make it easier to find later. Also find out how to print topics for reading offline.
- How do I view information in different languages?  
Find out how to view the information center in different languages.
- Where can I find other sources of help information?  
Find out how the documentation is structured, and what other sources of help are provided in the information center and the Message Brokers Toolkit.

## Where can I find other sources of help information?

New users: use this topic to find out about the resources available in the information center. You can also find out about other ways that you can get help using WebSphere Message Broker.

### What resources are available in the information center?

The Contents pane in the WebSphere Message Broker information center lists high level categories of information based on tasks that you might want to perform, such as configuring, administering, and developing applications. In each of the high level categories, concept topics provide definitions and background information to help you understand the tasks in the product. Task topics provide the actions or steps that you complete for the associated tasks. You can find reference topics in the Reference category that provide supporting information to help you complete the tasks, for example lists of options and parameters for commands.

Use the following links as alternate ways to help you find the information that you want:

- Where can I find a glossary of terms and abbreviations?  
The glossary contains a list of terms and abbreviations that are used in the WebSphere Message Broker information center.
- Where can I find the information center index?  
Search for information on particular subjects or terms using the Index.
- Where can I find support and technical documents on IBM Web sites?  
Search for WebSphere Message Broker support and technical documents on IBM Web sites. You must have an Internet connection to use the links on this page.
- Where can I find information about diagnostic messages?  
Search for information about diagnostic messages that might be generated when you use WebSphere Message Broker.

### What other sources of help are there in WebSphere Message Broker?

User assistance is provided in WebSphere Message Broker, in the form of messages, context-sensitive help on wizards and information provided on the product interfaces, including the Message Brokers Toolkit and in the command line environment.

- How do I get context sensitive help in the Message Brokers Toolkit?

Context-sensitive help and links into the information center are available in the Message Brokers Toolkit in the form of infopops. An infopop can be launched by selecting an object in the Message Brokers Toolkit and pressing the F1 key (on Windows) or SHIFT+F1 (on Linux).

- **How do I get assistance when I am developing my applications?**

You can get assistance when you develop your applications by using content assist. Content assist provides a list of available code options, for example the XPath functions to use in a message map, or a list of available references, for example to assist with constructing message references in ESQL. Content assist is available in the following editors for developing message broker applications:

- ESQL Editor
- Java Editor
- Message Mapping Editor

To access content assist, either select **Edit** → **Content Assist** or press Ctrl+Space to display a list of available options. To use content assist to help with constructing messages, you must have defined the message to the Message Brokers Toolkit and linked it to your message flow project; see “Project references” on page 66.

- **How do I get help when I am using commands?**

You can get usage information for the commands by entering the command without any parameters, or alternatively entering the command followed by /?, for example:

```
mqsilist /?
```

---

## Where can I get an overview of WebSphere Message Broker?

New users: use the links in this topic to get introductory information about WebSphere Message Broker.

- Where can I get a quick tour of the product?  
Learn about the basic concepts and features of WebSphere Message Broker with this animated tour.
- What is WebSphere Message Broker used for?  
Find out more about how WebSphere Message Broker is used in business.
- Where can I get a technical overview of the product?  
Learn about the capabilities and components of WebSphere Message Broker with this technical overview.
- Where can I find a WebSphere Message Broker example scenario?  
Learn how WebSphere Message Broker can be used to solve a business problem with this example scenario.

For more introductory information, see the IBM Redbooks publication WebSphere Message Broker Basics. This link works only if you have a connection to the Internet.

---

## How do I use the Message Brokers Toolkit?

New users: use the links in this topic to help you get started using the Message Brokers Toolkit. The Message Brokers Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform. You can use the Message Brokers Toolkit for developing, testing and debugging your applications. You can also configure and administer your broker domain using the Message Brokers Toolkit.

Some of the links in this topic work only if you are accessing the WebSphere Message Broker documentation from the Message Brokers Toolkit.

## How do I navigate and customize the Message Brokers Toolkit?

Use the links in this section to find out how to navigate the Message Brokers Toolkit. You can also find information about the different resources that you can create in the Message Brokers Toolkit.

- What are the perspectives in the Message Brokers Toolkit?  
A perspective defines the initial set and layout of windows in the workbench. Each perspective provides a set of functionality aimed at accomplishing a specific type of task or working with specific types of resources. Follow this link to find out about the Message Brokers Toolkit perspectives.
- What are the editors in the Message Brokers Toolkit?  
Follow this link to find out about the different editors in the Message Brokers Toolkit.
- What are the views in the Message Brokers Toolkit?  
Use this link to find out how views are used in the Eclipse workbench. You can use the following links to find out more about the views that are used in association with different editors in the Message Brokers Toolkit:
  - Navigator view
  - Tasks view (Problems and Alerts view)
  - Outline view
  - Properties view
- What types of resource can I create in the Message Brokers Toolkit?  
Find out about the types of resources and files that you can create and develop in the Message Brokers Toolkit.
- What are the keyboard shortcuts in the Message Brokers Toolkit?  
Find out how to navigate using keyboard shortcuts in the Message Brokers Toolkit.

## How can I customize the Eclipse workbench?

You can use the instructions in these links to learn how to use any application running in the Eclipse workbench, including the Message Brokers Toolkit and Rational Application Developer.

- How do I work with perspectives?  
Find out how to work with perspectives.
- How do I work with views and editors?  
Find out how to work with views and editors such as rearranging views and opening files for editing.
- How do I customize the workbench?

Find out how to work to further customize the Eclipse workbench such as rearranging the main toolbar and changing the placement of the tabs.

## How can I change the Message Brokers Toolkit preferences?

You can change a number of settings in the Message Brokers Toolkit to suit your requirements. Use these links to get more information about these settings and how to change them.

- What workbench preferences can I change?  
Use this link to find out about Message Brokers Toolkit preferences you can change and get links to information about preferences that might be useful when you develop message broker applications.
- How do I change the Broker Administration preferences?  
Follow this link to get an overview of how you can use the Broker Administration preferences to alter the behavior of the Broker Administration perspective. You might also want to change the Event Log editor preferences.
- How do I change the standard Eclipse preferences?  
Follow this link to find out about the standard preferences in the Eclipse workbench.

---

## How can I check that my installation was successful?

New users: use the links in this topic to help you check that your installation of WebSphere Message Broker was successful. The majority of links in this topic work only if you are accessing the WebSphere Message Broker documentation from the Message Brokers Toolkit.

- How do I create a default configuration?  
Use the Default Configuration wizard to create all the components you need to run the Message Broker samples. The default configuration also provides a useful environment to test and debug your own message flow applications. You can run a wizard to remove the default configuration when you have finished with it.
- Where can I find sample applications?  
The Message Broker samples are a set of sample message flow applications and associated resources that demonstrate some of the capabilities of WebSphere Message Broker. Use this link to view a description of all of the samples. You can also follow this link to access the Samples Gallery. You can view samples only when you use the information center that is integrated with the Message Brokers Toolkit. You must have created the default configuration to run the samples.  
You can use the following samples to quickly check that your installation was successful:
  - Pager samples
  - Scribble sample
  - Soccer sample

More information about checking that your installation of WebSphere Message Broker was successful, can be found in the Installation Guide.



---

## How do I design and develop applications?

New users: use the links in this topic to get information about how to design and develop message flow applications and related resources.

- “What do I need to know to start developing applications?”

Use this link to get information that it is useful to know before you start developing your applications.

- How do I design a message flow?

Follow this link to get a comprehensive list of the decisions that you need to make when designing a message flow.

- “How do I construct message flows?” on page 12

Find out how to construct message flows.

- “How do I program message flows?” on page 14

Find out how to program your message flow nodes using ESQL, Java, message mappings and XML transformations.

- “How do I design a message model?” on page 18

Find out how to design message models.

- “How do I develop a message model?” on page 19

Find out how to develop message models.

- “How do I develop publish/subscribe applications?” on page 21

Find out how to design and develop publish/subscribe applications.

- How do I develop user-defined extensions?

Find out how to develop your own user-defined extensions such as nodes and parsers.

## What do I need to know to start developing applications?

New users: use the links in this topic to get information about concepts that are useful to understand before you start developing your applications. This topic contains the following sections:

- “Basic application development concepts”
- “Advanced application development concepts” on page 11

## Basic application development concepts

### What is WebSphere Message Broker typically used for?

Use the links in this section to find out about the tasks that WebSphere Message Broker is commonly used for. You can also find out about the alternative ways to implement message flows for these tasks.

- **How can I use WebSphere Message Broker to route messages?**

You can select from a number of ways of routing messages using message flows:

- Using a Filter node

You can route messages through a message flow based on the content of the message using a Filter node. The nodes that are connected to the different terminals of the Filter node can be used to apply different processing and routing to messages based on their content.

- Using the destination list to route messages

You can create a destination list to define the recipients of output messages using a Compute node. This list can then be used to route messages using a

RouteToLabel and Label nodes. Alternatively a single message can be sent to many locations using a destination list for the destination mode, on some output nodes.

- Topics

In publish/subscribe messaging, a message can be assigned a topic by a publisher. The broker matches the topic with a list of applications that have subscribed to the matched topic and delivers the message to each of the subscribing applications.

You can see examples of how message flows can be used for routing in the following samples. The links to the samples work only if you are accessing the WebSphere Message Broker documentation from the Message Brokers Toolkit.

- Message Routing sample
- Airline Reservations sample

- **How can I use WebSphere Message Broker to transform and enrich messages?**

You can select from a number of different ways of transforming and enriching messages using different nodes in message flows. Messages can be enriched with content from databases, applications and files. The following methods of programming nodes in your message flows can be used to transform messages:

- ESQL

Extended Structured Query Language (ESQL) is a programming language based on Structured Query Language (SQL) that is commonly used to work with databases. ESQL is extended to access and manipulate data in messages and databases.

- Java

The JavaCompute node can be used to add any valid Java code to your message flow. You can access, create and transform messages using Java in your message flows.

- Message mappings

Message mappings use a drag-and-drop interface to transform messages. You can use conditional logic, ESQL functions and XPath functions to create complex mappings and transformations. You can also use message mapping to select and manipulate data in database tables.

- XML transformation

You can use the XMLTransformation node to transform from one type of XML message to another based on rules provided by an eXtensible Stylesheet Language (XSL) stylesheet.

The Airline Reservations sample demonstrates message enrichment, and the following samples demonstrate message transformation. The links to the samples work only if you are accessing the WebSphere Message Broker documentation from the Message Brokers Toolkit.

- Video Rental sample
- XMLT sample

## **What types of application can I develop?**

You can develop the following kinds of applications:

- Message flows

A message flow is a sequence of processing steps that run in the broker when an input message is received.

- Message models

You can define the structure of messages for use with your message flows.

- Publish/Subscribe

Publish/subscribe is a style of messaging in which the providers of information (publishers) are decoupled from the consumers of that information (subscribers).

- User-defined extensions overview

A user-defined extension is an optional component that is designed by the user to extend the functionality of WebSphere Message Broker. A user-defined extension can be either a node or a parser.

- Configuration Manager Proxy applications

The Configuration Manager Proxy (CMP) is an application programming interface that you can program to control broker domains using a remote interface to the Configuration Manager.

## Advanced application development concepts

### How can I connect my applications?

Use the links in this section to find out how you can connect your applications to WebSphere Message Broker.

- What are the application communication models?

Find out about the differences between the two types of application communication model that WebSphere Message Broker supports.

- What application programming interfaces are supported?

Find out about the many programming interfaces supported by WebSphere Message Broker.

- What application transports are supported?

Find out about the communication transports and protocols that WebSphere Message Broker supports.

### What is a logical message structure?

Use the links in this section to find out about the logical message structure and the internal representation of messages within message flows. When a message is received by a message flow a logical structure is created called the logical tree. The logical tree contains a message tree, which is the internal representation of the physical message, plus a number of other trees that are used to store information during the processing of the message in the message flow.

- What is the logical tree structure?

Use this link to get more information about the logical tree structure and how it is used in application development. You can get information about the four subtrees from the following links:

- Message tree structure
- Environment tree structure
- LocalEnvironment tree structure
- ExceptionList tree structure

- How is the message tree populated?

Follow this link to get detailed information on how the message tree is populated when a message enters a message flow, and how the structure of the message tree is affected by the transport protocol that the message is received on.

## What is a physical message structure?

Use the links in this section to get information about the physical messages flowing through a system, and to find out how the different structures and formats are handled by your WebSphere Message Broker applications.

- What are the differences between predefined and self-defining messages?  
Find out about the differences between predefined and self-defining messages in WebSphere Message Broker. The design of your applications can be influenced by your decision to use predefined or self-defining messages. If you use predefined messages you must create a message model to define the logical message structure to be used by the nodes in your message flows, but this is optional if you use self-defining messages.
- Why model messages?  
Find out about the benefits of modeling messages, even if you use self-defining messages in your system.
- What can I find an introduction to message modelling?  
Follow this link to get introductory information about message modelling.
- How can I define a physical format for my message?  
Follow this link to find out how you can define the physical structure for your messages. The following links provide information on specific types of message format:
  - What is a Custom Wire Format?
  - What is an XML Wire Format?
  - What is a Tagged/Delimited String Format?
- Which message domain and format should I use?  
A message model has one or more domains that determine which parser is used to construct the logical tree structure from the physical message. Follow this link for guidance on selecting a domain for your message model.
- What is a parser?  
A parser is used to interpret the physical structure of an incoming message and create an internal representation of the message in a tree structure - the logical message tree. Follow this link to get an introduction to parsers.
- Which parser should I use?  
Follow this link to get assistance on choosing which parser to use in your message broker applications.

## How do I construct message flows?

New users: when you have considered the various factors involved in designing a message flow, you are ready to create one. Use the links in this topic to find out how to construct message flows and work with related resources.

### How do I create message flow resources?

- How do I create a message flow project?  
A message flow project is a specialized container in which you create and maintain all the resources associated with one or more message flows. You can group together related message flows and resources in a single message flow project to provide an organizational structure to your message flow resources. You must create a project before you can create a message flow. Use this link to find out how to create a message flow project.
- How do I create a broker schema?

If you want to organize your message flow project resources, and to define the scope of resource names to ensure uniqueness, you can create broker schemas. When you first create a message flow project, a default broker schema is created within the project. Use this link to find out how to create a broker schema.

- How do I create a message flow?

You create a message flow to specify how to process messages in the broker. You can create any number of message flows and deploy them to one or more brokers. Use this link to find out how to create a message flow.

## How do I construct a message flow?

- How do I add a node?

When you create a new message flow, the first action to take to define its function is to add nodes. A message flow node is a processing step in a message flow. A message flow node can be a built-in node, a user-defined node, or a subflow node. Select the nodes to add to your message flow from the message flow node palette.

- How do I rename a message flow node?

To make your message flows easier to understand and to maintain, you can change the name of any type of node that you have added to your message flow. For example, you might change the name of an MQInput node to match the input queue name defined in the node.

- How do I configure a node?

When you have included an instance of a node in your message flow, you can configure it to customize its function. Each node has a set of properties that are specific to that node's function. Built-in nodes lists the nodes that are provided with WebSphere Message Broker, arranged by category. Select the required node from this list to view details of the terminals, and properties, and how to configure the node. "How do I program message flows?" on page 14 contains more information about creating and configuring resources for nodes that can be programmed, such as the JavaCompute, Compute, and Mapping nodes.

- How do I connect nodes?

You must connect the nodes in your message flow to indicate how the flow of control passes from input to output, and the route that messages can take through the message flow. A message flow node has a fixed number of input and output points known as terminals. You can connect the failure or catch terminals of nodes to add error handling to your message flows.

- How do I align and arrange nodes in a message flow?

You can change the way that the nodes in your message flow are displayed and arranged, to make them easier to read. You can add a bend point to make the flow of control easier to follow where node connections cross each other.

- How do I save a message flow?

Save the message flow when you want to:

- Close the workbench
- Work with another resource
- Validate the contents of the message flow

## How do I manage my message flows?

- How do I add a subflow?

After you have created message flows, you can embed them in other message flows as subflows. You can use subflows to reuse function across message flow projects, reduce development time, and increase the maintainability of your message flows.

- How do I remove a message flow node?  
Find out how to remove a node from your message flow.
- How do I remove a node connection?  
Find out how to remove a node connection between nodes in your message flow.
- How do I delete a message flow?  
Find out how to delete your message flow.
- How do I move a message flow?  
Find out how to move your message flow between broker schemas or message flow projects.
- How do I rename a message flow?  
Find out how to rename your message flow.
- How do I delete a message flow project?  
Find out how to delete your message flow project.
- How do I back up my message flow projects and related resources?  
Your message flow projects and other resources are stored in the Message Brokers Toolkit workspace and other locations in your file system. You can use the Export function in Eclipse, or take copies of the directories on your file system, to back up your resources. Alternatively, you can set up the Message Brokers Toolkit to work with a “Development repository” on page 66.

## How do I program message flows?

New users: use the links in this topic to get information on the concepts and tasks associated with programming message flows. This topic contains the following sections:

- “How do I use message mappings?”
- “How do I use ESQL?” on page 15
- “How do I use Java?” on page 16
- “How do I use XML transformations?” on page 17

## How do I use message mappings?

Message mappings use a drag-and-drop interface to create and transform messages. You can use conditional logic, ESQL functions, and XPath functions to create complex mappings and transformations. You can also use message mapping to select and manipulate data in database tables. You must create a message definition for any messages that you include in a message mapping. Message map files are stored in message flow projects.

- How do I create a message map file?  
Get instructions on how to create a message map from the Broker Development view. You can also create a message map from the nodes that support message mapping including:
  - Mapping node
  - DataInsert node
  - DataUpdate node
  - DataDelete node

- What does the Message Mapping editor look like?

You create and modify message mappings in the Message Mapping editor. Use [this link](#) to find out about the Message Mapping editor.

- How do I configure message mappings?

When you configure message mappings you can drag content from a source to a target. The source can be a message, a database, or both, and the target can be messages, database tables, or both. If your target is a database, you can select the database operation insert, update, or delete that you want to perform on the table. You can set the value for your target to be a constant, or you can use a function or expression to produce the value. Additionally, you can configure conditional mappings to set different values for targets based on the content of the sources, and to handle repeating elements in sources and targets. Follow [this link](#) to find out more about message mappings.

- **Message mapping syntax**

When you use an expression to set the value of a target in a message map, the expression must be in XPath format.

- What syntax is used in mapping nodes?

Use [this link](#) to find out about the XPath syntax used in message maps.

- How do I use expressions in my message maps?

Use [this link](#) to find out about the type of functions that you can use in expressions in your message maps.

- How can I find out more about XPath query syntax?

To find out more about XPath, follow [this link](#) to the W3C recommended XPath 1.0 query syntax reference document. This link works only if you have an active internet connection.

- How do I create and call submaps and routines?

You can reuse message maps between different message flows and message flow projects using a submap. You can also use a submap to create message mappings for a wildcard source so that you can select replacement elements, for example to select the appropriate body element from a SOAP message. In addition to calling a submap from a message map, you can call a submap from ESQL. You can also call ESQL routines from a message map. Use [this link](#) to find more information about creating and using submaps and ESQL routines.

## How do I use ESQL?

Extended Structured Query Language (ESQL) is a programming language based on Structured Query Language (SQL), which is commonly used with relational databases such as DB2. ESQL extends the constructs of the SQL language to provide support for you to work with both message and database content. ESQL can be used with the Compute, Database, and Filter nodes. Many of the WebSphere Message Broker Samples demonstrate how to use ESQL in message flows. You can view samples only when you use the information center that is integrated with the Message Brokers Toolkit. The following samples contain example ESQL code:

- Airline Reservations sample
- Large Messaging sample
- Message Routing sample

Use the following links to find out how to use ESQL in your message flows.

- How do I create an ESQL file?



The ESQL code for each of your nodes is contained within a module in an ESQL file. Use this topic to find out how to create an empty ESQL file in your message flow project. Alternatively, you can select **Open ESQL** to create an ESQL file when you configure the first node in a message flow that uses ESQL. This action creates an ESQL file with skeleton ESQL code for a module associated with the selected node. You can also select an ESQL module for a node from a different message flow project by creating “Project references” on page 66.

- What is the ESQL editor?

You can create and modify your ESQL code in the ESQL editor. You can change the way that code is displayed in the ESQL editor, and modify the way in which the ESQL editor validates code, by changing your ESQL preferences.

- How do I create ESQL for a node?

Follow this link to get an overview of how to create ESQL for your node, including information about the differences between the skeleton ESQL code generated for the modules associated with the Compute, Database, and Filter nodes.

- How do I modify ESQL for a node?

Follow this link to find out how to modify the skeleton ESQL module code.

- How do I save an ESQL file?

Find out how to save your ESQL file.

- How do I write ESQL code?

Follow this link to get introductory information about writing ESQL code for your message flows. Find out how to manipulate the message tree, transform data, access databases, and work with messages from different message domains using ESQL.

- **ESQL language**

Follow the links below to get concept and reference information about the ESQL language:

- ESQL data types
- ESQL variables
- ESQL operators
- ESQL field references
- ESQL statements
- ESQL functions
- ESQL procedures
- Special characters, case sensitivity, and comments in ESQL
- ESQL modules
- ESQL reserved keywords
- ESQL non-reserved keywords

## How do I use Java?

You can create a Java class file for a JavaCompute node and code Java functions to tailor the behavior of the node. You can add any valid Java code to your JavaCompute nodes and use the Java user-defined node API to process messages. Java files are managed through the Java perspective.

The JavaCompute Node sample provides a collection of message flows that demonstrate how to use the JavaCompute node. You can view samples only when you use the information center that is integrated with the Message Brokers Toolkit.



- How do I use a JavaCompute node?  
You can use the JavaCompute node to examine the content of an input message, transform a message, and build new messages. Follow this link to find out how to use and configure the JavaCompute node in your message flows.
- How do I create Java code for a JavaCompute node?  
Find out how to create a Java class file using the JavaCompute node wizard. The JavaCompute node wizard creates a Java class with skeleton Java code that is based on the options that you select in the wizard. You can then modify the skeleton code to perform your own processing.
- How do I open an existing Java file?  
You can modify existing Java code that you have created in a Java Project.
- How do I save a Java file?  
Find out how to save your Java file.
- How do I write Java for message flow applications?  
Get introductory information about writing Java code for your message flow applications, including how to manipulate the message tree, access databases, handle errors, and access broker properties.
- What views and editors do I use when programming Java?  
Get a list of concept, task, and reference topics that relate to editors and views for working with Java. This link works only if you are accessing the information center from the Message Brokers Toolkit.
- Where can I get assistance when programming Java?  
You can use code assist to provide a list of available command completions that you can select to insert into the editor. You can also use command assist to access Javadoc information about code in the Java editor. This link works only if you are accessing the information center from the Message Brokers Toolkit.
- How do I add Java code dependencies?  
Find out how to include references to other Java projects and JAR files in your JavaCompute node code.
- Where can I find the Java user-defined node API?  
Follow this link to the Java API for the WebSphere Message Broker classes for creating a Java user-defined node, which you can also use to code your JavaCompute node.

## How do I use XML transformations?

You can use the XMLTransformation node to transform an XML message into another form of XML according to the rules provided by an XSL (eXtensible Stylesheet Language) stylesheet. You can specify the location of the stylesheet to apply to this transformation in three ways:

- Use node properties, which ensures that the transformation that is defined by this single stylesheet is applied to every message that is processed by this node.
- Use the content of the XML data within the message itself, which transforms the message according to a stylesheet that the message itself defines.
- Set a value within the LocalEnvironment folder.

You can find links to information on using XML transformation in this section.

- Where can I find an sample application that demonstrates XML transformations?  
If you have the Message Brokers Toolkit installed, you can use this link to display the XMLT sample in the Samples Gallery. The XMLT sample is a

message flow sample application that shows to use a message flow to transform an XML message to another form of XML message according to the rules provided by an XSL stylesheet.

- How do I use the XMLTransformation node?

Follow this link to find out how to use and configure the XMLTransformation node to transform an XML message into another form of XML

- Where can I find out more about XSL Transformations?

Find out more about XML transformation using this link to the W3C specification of the syntax and semantics of the XSL Transformations language for transforming XML documents into other XML documents.

## How do I design a message model?

New users: use the links in this topic to find out how to design a message model.

If the format of the messages that you want to use with your applications is not self-defining, then you must create a message model that defines the structure of your messages. If your messages are self-defining, you might want to create a message model to take advantage of runtime validation of messages, reuse of messages, automatic generation of documentation, and code completion on the message structure when you use ESQL. If you want to use message mapping, you must also create a message model for your messages.

WebSphere Message Broker provides pre-built models for common industry standard message formats such as SWIFT, EDIFACT, X12, FIX, HL7 and TLOG. You can also create message models from C header files, COBOL copybooks, XML Schema and DTDs, and WSDL files. Alternatively, you can use the Message Definition editor to create your own message models.

For further information, read the logical and physical message structure sections in “What do I need to know to start developing applications?” on page 9

- What is the structure of a message tree?

The message tree is a part of the logical message tree in which the broker stores its internal representation of the message body. Follow this link to find out more about the message tree.

- How is the physical format of a message defined?

Find out how the MRM domain can be used to define the physical format of your messages.

- How can I create message definitions from existing formats?

You can populate your message set with message definitions that are created from existing application message formats. Use this link to find out about each of the formats that you can import to create message definitions. You can also get links to information about the supported features for each message format from Import formats.

- **Message Model Components**

Use the following links to find out about the different components that are used to build a message model:

- What is a message set project?

A message set project is a container in which you create and maintain all the resources that are associated with a single message set.

- What is a message set?

A message set is a logical grouping of messages and the objects that comprise them (elements, types and groups). A message set can contain one message set file, definition files and message category files.

- What is a message definition file?

A message definition file contains the messages, elements, types, and messages that make up a message set. The message definition file contains the logical model and associated physical model in XML Schema form for a group of related messages.

- What is a message model object?

Get an overview of the objects that make up a message model.

- What is a message category?

You can use message categories to group your messages or to assist in the generation of Web Services Description Language (WSDL) files.

- What is a multipart message?

You can define a message that contains one or more embedded messages within its structure.

- How can I control the number of occurrences of an element or attribute in a message?

You can define a message that contains repeating, optional and mandatory elements.

- What are substitution groups?

You can create substitution groups for XML-format messages, where one global element can be replaced by another global element in the message.

- How are namespaces used in the message model?

Objects in the message model such as elements, attributes, types, and groups are identified by their name. No two objects in the same scope are allowed to have the same name. If namespaces are enabled for a message set, each message definition file within it can specify a namespace. Global objects within namespaces can share the same name, therefore namespaces provide a way to avoid name clashes between objects.

## How do I develop a message model?

New users: use this topic to get information about how to create message models.

The Video Rental sample provides step by step instructions on how to create a simple message model. The sample also demonstrates message transformation between three different message formats. You can view samples only when you use the information center that is integrated with the Message Brokers Toolkit.

## How do I create and configure a message model?

Use the links in this section to find out how to create and configure message set projects and message set files.

- How do I create a message set?

You must create a message set before you can add any content to your message model. When you create a message set, a message set project is also created.

- **How do I configure a message set file?**

After you have created a message set, you must configure the logical and physical properties of the message set. You can use the Message Set editor to configure the properties of a message set.

- How do I configure the logical properties of my message set?

- Find out how to configure logical properties of a message set.
- How do I configure the physical formats of my message set?  
Find out how to add and configure different physical format layers in a message set.
- How do I configure the properties of my message set?  
Get information about general message set properties and links to specific information for the different physical format layers.

- How do I create a message definition file?

You must create a message definition file before you can create the message model objects. You can use the New message definition file wizard to create an empty message definition file, or you can populate a message definition using existing application message formats by Importing data structures.

- How do I configure the properties of my message definition files?

When you have created a message definition file, you can configure the message definition file properties described in this topic using the “Message Definition editor” on page 98.

- **How do I Add and configure message model objects?**

When you have created a message definition file you can add message model objects to your message definition file, to define the structure of your message. You can also add objects to existing message definition files. You must then configure the logical, physical and other properties of the message model objects.

- How do I add message model objects?

Follow this link to find out how to add different message model objects to a message definition file.

- How do I configure message model objects?

Follow this link to find out how to configure message model objects.

- What properties do message model objects have?

Follow this link to get reference information about logical, physical and documentation properties for all types of message model objects.

- How do I create a multipart message?

You can create a message model that includes a multipart (embedded) message. Use this link to find out how to create a multipart message.

- How do I link from one message definition file to another?

Follow this link to find out how to link one message definition file to another.

## How do I manage my message models?

Find out about how to manage your message models:

- **How do I work with message category files?**

You can create a message category file to organize your messages within your message set or to assist in the generation of Web Services Description Language (WSDL) files.

- How do I create a message category file?

Find out how to create a message category file.

- How do I add a message to a message category?

Find out how to add a message to your message category files.

- How do I configure message category file properties?

Find out how to view and modify the properties of your message category files in the “Message Category editor” on page 97. You can edit the following properties for the message category file:

- Message category properties
- Message category member properties

- **How do I generate model representations from message models?**

When you have created and populated a message set, you can generate a message model in different representations for use both by WebSphere Message Broker and your applications. You can also generate documentation in the form of HTML files that can be used by business analysts or developers to understand the logical format of your messages. Use the following links to find out about the types of model representations that you can generate from your message models:

- Generate message dictionaries
- Generate XML Schema
- Generate WSDL
- Generating message set documentation as HTML

- How do I configure message set preferences?

Follow this link to find out how to make changes to preferences that relate to the handling of message set processing.

- How do I change my viewing preferences for message model editors?

You can change your viewing preferences for the Message Set editor and the Message Definition editor, to make viewing message definition files easier.

## **How do I develop publish/subscribe applications?**

New users: use the links in this topic to get concept and reference information that is useful when you develop publish/subscribe applications. Publish/subscribe is a style of messaging application in which the providers of information (publishers) are decoupled from the consumers of that information (subscribers) using a broker. The following samples are examples of simple publish/subscribe message flows, and include associated end user applications. You can view samples only when you use the information center that is integrated with the Message Brokers Toolkit.

- Pager samples
- Scribble sample
- Soccer sample

A publish/subscribe application can consist of a network of brokers connected together. By connecting brokers together, publications can be received by a client on any broker in the network. A publish/subscribe topology consists of the brokers, the collectives, and the connections between them, that support publish/subscribe applications in the broker domain. Links to information about setting up a publish/subscribe topology and associated administration tasks are provided in “How do I plan and configure a broker domain?” on page 24

- Where can I find an overview of publish/subscribe?

Get an overview of publish/subscribe. Use the following links for more information about the components of a publish/subscribe system:

- Topics
- Publishers
- Publications
- Subscription points

- Subscribers
- Subscriptions

- What is publishing?

An application can publish information that is then sent by the broker to all interested subscribers.

- What are retained publications?

By default, a broker discards a publication after it has sent that publication to all interested subscribers. However, a publisher can specify that it wants the broker to keep a copy of a publication, which is then called a retained publication.

- How do applications subscribe to a topic?

Find out how applications subscribe to receive messages of interest.

- How do I deregister a subscription?

If an application is no longer interested in receiving certain publications, the application can deregister its subscription.

- How do I use a publication node?

Use the Publication node in your message flows to filter output messages and transmit them to subscribers who have registered an interest in a particular set of topics.

- What are command messages?

Different types of command messages can be sent from an application to instruct the broker to perform the required task; for example to publish a message, or register a subscriber. Applications can also receive messages from which might include Reason codes if a problem occurred when completing the publish/subscribe request.

## How do I deploy and test applications?

New users: use this topic to get information about deploying and testing your applications during development. When you have developed your message flows, message models and other resources, you are ready to deploy them to a broker. You can use features provided in the Message Brokers Toolkit to help you test your applications. Read “How do I deploy and configure applications?” on page 27 for information about deploying and configuring applications for use in a production environment.

In order to deploy and test your applications you must have a broker domain.

You can Create the Default Configuration to create a broker domain suitable for testing applications during development. This link works only if you are accessing the information center from the Message Brokers Toolkit.

## How do I deploy message flow applications and related resources?

Use the links in this section to get instructions on how to deploy your message flows, message sets and related resources to a broker. Read Message flow application deployment for an overview of the concepts of deployment of application resources.

- How do I create a server project?

You must have a server project to create a broker archive file. Follow this link for instructions on how to create a server project. Alternatively, you can use the server project that contains your broker domain configuration details.



- How do I create a broker archive?

A broker archive file is a compressed file that contains compiled message flows, message dictionaries and other application development resources. Use this link to get instructions on how to create a broker archive. You can use the “Broker Archive editor” on page 91 to add or remove message flows and message sets to, or from, your broker archive. You can also use the Broker Archive editor to edit the configurable properties within your broker archive.

- How do I add files to a broker archive?

You can add message flows and message sets to a broker archive file. The message flows, message sets and related resources are compiled when they are added to the broker archive file. You can also choose to add the source files to the broker archive file. You can not add resources to a broker archive file from a project that contains an error.

- How do I deploy a broker archive?

Find out about the different ways that you can deploy a broker archive file to a broker. For deploying applications for development to a test environment, the simplest method is to use the Message Brokers Toolkit.

- How do I check the results of the deployment?

Find out about the ways that you can check the results of deployment. When you deploy applications to a test environment, the simplest method is to use the Message Brokers Toolkit. You can also use the mqsilist command with the name of the broker and execution group to which you deployed your broker archive file to.

- How do I edit configurable properties?

System objects that are defined in message flows have configurable properties that you can update within the broker archive before deployment. By changing configurable properties, you can customize a broker archive for a new domain, without needing to edit and rebuild the message flows or other application resources.

- How do I refresh the contents of a broker archive?

You can refresh the contents of a broker archive by removing resources from it and, having made required changes, add them back again. Alternatively you can use the Refresh option in the Broker Archive editor.

## How can I test my applications?

Use the links in this section provide information about the different features provided in the Message Brokers Toolkit to help you to test your applications.

- How do I test a message flow using the Test Client?

You can test message flows that use WebSphere MQ or HTTP input and output nodes. If you have a message set defined for the input nodes in your message flow, then the message flow test tool can provide an input message template in XML format, that you can use to develop test messages.

- How can I put a test message onto a WebSphere MQ queue?

Get instructions on how to create and use an enqueue file to put a test message onto a WebSphere MQ queue.

- How can I get a test message from a WebSphere MQ queue?

Get instructions on how to read a message from a WebSphere MQ queue using the Dequeue Message dialog.

- How can I use the flow debugger to help test my message flow applications?

You can use the flow debugger to help test your message flows by tracking the path a message takes through your message flow. You can also step through the ESQL, Java, and message mappings in your message flows, to view the output messages being constructed.

- “How can I diagnose problems?” on page 31

If you are having difficulty deploying your message flows and other application resources, or receiving unexpected results when you test your message flows, take a look at the problem determination information in this topic.

---

## WebSphere Message Broker Administration

New users: use the links in this topic to get information on how to plan, configure and administer your broker domain.

- “How do I plan and configure a broker domain?”

Get information about how to plan and configure a broker domain.

- “How do I deploy and configure applications?” on page 27

Get information about how to configure and deploy your applications.

- “How do I administer and monitor a broker domain?” on page 30

Get information on day to day administrative tasks for WebSphere Message Broker. Also use this link to find out how to monitor the performance of your applications.

### How do I plan and configure a broker domain?

New users: use the links in this topic to find out how to plan and configure a broker domain. You can create a simple broker domain for testing purposes on your Message Brokers Toolkit machine using the Default Configuration wizard. When you are ready to create a broker domain for testing on platforms other than Windows or Linux (x86 platform), or for production purposes, you must plan your broker domain carefully using the information provided in the WebSphere Message Broker information center.

### What do I need to know about broker domains?

A broker domain is one or more brokers that share a common configuration, together with the single Configuration Manager that controls them. Use the links in this section to find information that you might find useful before planning and designing your broker domain.

- What is a broker?

A broker is a set of execution processes that hosts one or more message flows to route, transform, and enrich inflight messages. Follow this link to get information about brokers, and the resources that you need to create and configure your brokers.

- What is a Configuration Manager?

The Configuration Manager is the interface between the workbench and an executing set of brokers. It provides brokers with their initial configuration, and updates them with any subsequent changes. It maintains the broker domain configuration. The Configuration Manager is supported on all the broker platforms. Follow this link to get information about the main functions of the Configuration Manager, and the resources that you need to create and configure your Configuration Manager.

- What is a User Name Server?



The User Name Server is an optional runtime component that provides authentication of users and groups performing publish/subscribe operations. Follow this link to get information about the User Name Server.

- What is a publish/subscribe topology?

A publish/subscribe topology consists of the brokers, the Collectives, and the connections between them, that support publish/subscribe applications in the broker domain. Follow this link to get an introduction to broker networks and publish/subscribe topologies.

- What do I need to know about databases?

WebSphere Message Broker uses databases to store internal data about the broker. You must create and configure the broker database before you can create any brokers. Follow this link to find out more about broker databases and how to create connections to them.

## What do I need to know about authorization and security?

Each broker domain must contain one Configuration Manager, which manages the configuration of the domain, and at least one broker. The number of brokers that you include in the broker domain depends on several factors, including your performance requirements and the type of applications in your broker domain. If the broker domain is for use with publish/subscribe applications, you can include a User Name Server to manage access to topics. Follow the links in this section to get information about authorization, access control and security for your broker domain.

- What authorizations are needed for configuration tasks?

Authorization is the process of granting or denying access to a system resource. Use this link to get examples of the tasks in WebSphere Message Broker that require authorization. Use the following links to get more details about the security requirements for administrative tasks on the different platforms:

- Security requirements for Linux and UNIX platforms
- Security requirements for Windows platforms
- Security requirements for z/OS

- What authorizations are required for access to runtime resources?

Follow this link to find out how Access Control List (ACL) entries and groups are used to authorize access to broker domain components.

- How are Access Control Lists used?

Access Control List (ACL) entries allow or deny a user to access an object. Follow this link to get examples of how Access Control List entries are used to govern which users can manipulate objects in a broker domain. You can get more information about Access Control List permissions from ACL permissions.

- Where can I find out about security for publish/subscribe?

Follow this link to get an overview of publish/subscribe security.

## How do I plan a broker domain and configure prerequisites?

Use the following links to find out how to plan your broker domain. You can also find out about tasks you must complete before you can configure your broker domain.

- How do I plan a broker domain?

Follow this link to find out what you need to consider when you plan your broker domain.

- How do I set up broker domain security?

Follow this link for information on how to set up security for your broker domain. Each link comprises a list of reminders or questions about the security tasks to consider for your broker configuration. The answers to the questions provide the security information that you need to configure your WebSphere Message Broker components, and also give you information about other security controls that you might want to deploy.

- How do I configure my databases?

WebSphere Message Broker uses databases for two purposes:

- Component databases, which the broker uses to store its internal data. Each broker must have its own set of tables in the database but multiple brokers can share the same database.
- User databases, which you use to store your business data that can be processed by your message flow applications.

You must manually create and configure the broker database or databases using a supported database manager before you can create your brokers. When you create your brokers, the tables for each broker are automatically created in the database. Follow this link to get instructions on how to create and configure databases for use as broker database, or a user database.

## How do I create and configure broker domain components?

A broker domain must contain at least one broker and the Configuration Manager. If the broker and Configuration Manager share the same queue manager, you do not need to create any WebSphere MQ infrastructure to connect them together. In more complex broker domains, you might create several brokers (each on their own queue manager), a Configuration Manager, and WebSphere MQ channels (or a WebSphere MQ cluster) to connect the components together. You might also create several execution groups on each broker to manage the message flow applications that you deploy to the broker. If you are creating a broker domain for publish/subscribe applications, you can optionally create a User Name Server to manage publisher and subscriber authorities. To create a broker domain, complete the following tasks in order:

1. How do I create a broker?

On Windows and Linux (x86 platform), you can create brokers in the Message Brokers Toolkit using the Command Assistant wizard. Alternatively, or on UNIX and z/OS platforms, create the components from the command line. Create as many brokers as you require.

2. How do I create the Configuration Manager?

On Windows and Linux (x86 platform), you can create the Configuration Manager in the Message Brokers Toolkit using the Command Assistant wizard. Alternatively, or on UNIX and z/OS platforms, create the components from the command line. Create only one Configuration Manager for each broker domain.

3. Optional: How do I create the User Name Server?

On Windows and Linux (x86 platform), you can create the User Name Server in the Message Brokers Toolkit using the Command Assistant wizard. Alternatively, or on UNIX and z/OS platforms, create the components from the command line. Create only one User Name Server per broker domain, and only if you want to manage access to publish/subscribe applications.

4. How do I verify that all the components have been created successfully?

You can use the `mqsilist` command to check that your components were created successfully

| 5. How do I connect the components together?.

| If you have created a single broker that shares a queue manager with the  
| Configuration Manager, the broker and Configuration Manager can  
| communicate without any additional configuration. For components that are  
| on different queue managers to communicate, however, you must create  
| WebSphere MQ channels between the queue managers. You can do this by  
| manually creating all the channels or by defining the queue managers in a  
| cluster. For more information about queue manager clusters, see the  
| WebSphere MQ documentation.

| 6. How do I start the Configuration Manager?

| Follow this link to get instructions on how to start the Configuration Manager.

| 7. How do I start a broker?

| Follow this link to get instructions on how to start a broker.

| 8. How do I start the User Name Server?

| Follow this link to get instructions on how to start the User Name Server.

| 9. How do I create a domain connection?

| You can configure your broker domain using the Message Brokers Toolkit on  
| Windows and Linux (x86 platform). You to configure and administer your  
| broker domain components using the “Broker Administration perspective” on  
| page 81 in the Message Brokers Toolkit.

| 10. How do I add a broker to my broker domain?

| Adding a broker to the broker topology creates a reference to the broker in the  
| configuration repository. The physical broker must have been created before  
| you can add the broker using the Message Brokers Toolkit. Follow this link to  
| find out how to add a broker to your broker domain. When you add a broker  
| to the broker domain, the default execution group is added to the broker,  
| ready for message flows and related resources to be deployed to it.

## | **How do I deploy and configure applications?**

| New users: use the links in this topic to get information on deploying and  
| configuring message flow applications and related resources.

### | **What do I need to know about deployment?**

| Deployment is the process of transferring data to an execution group belonging to  
| a broker, so that it can take effect in the broker domain. Use the following links to  
| find out more about deployment.

- Where can I get an overview of deployment?

| Use this link to find out about the Message Brokers Toolkit, the mqsideploy  
| command and the Configuration Manager Proxy API as environments for  
| deployment.

- What do I need to know about message flow application deployment?

| Use this link to find out more about deploying message flow applications and  
| related resources.

- What is a broker archive?

| You add message flow applications and related resources to a message broker  
| archive in order to deploy the resources to an execution group. Use this link to  
| find out about the files that can be associated with a broker archive file.

- What are configurable properties of broker archives?

| System objects that are defined in message flows can have configurable  
| properties that you can update within the broker archive file before deployment.

You can use configurable properties to update target-dependent properties, such as queue names, queue manager names, and database connections.

- How do I view version and keyword information for deployable resources?  
Follow this link to find out how to view the version and keyword information of deployable objects. You can use version and keyword information to track resources that have been deployed to a broker.
- How do I deploy changes to my broker configuration?  
You can use a broker configuration deployment to inform a broker of changes you have made to configuration settings.
- How do I deploy changes to my publish/subscribe topology?  
Topology deployment is used with publish/subscribe. It informs each broker in the domain of the brokers with which it can share publications and subscriptions.
- How do I deploy changes to my publish/subscribe topics hierarchy?  
Topics hierarchy deployment is used with publish/subscribe. It is used to communicate a modified topics hierarchy or to inform a new broker of the topics hierarchy.

## How do I deploy message flow applications?

You must have a broker domain configured and the components running to deploy message flow applications to an execution group. Use the links in this section to find out how to deploy your message flow applications and related resources to an execution group.

- **How do I add an execution group?**  
An execution group is a named grouping of message flows that have been assigned to a broker. The broker enforces a degree of isolation between message flows in distinct execution groups by ensuring that they execute in separate address spaces, or as unique processes. When you add a broker to a broker domain, the default execution group is created automatically. You can create more execution groups in a number of different ways:
  - Using the Message Brokers Toolkit
  - Using the command line
  - Adding an execution group on z/OS
- How do I create a server project?  
You must have a server project to create a broker archive file. Use this link to get instructions on how to create a server project. Alternatively, if you have created a default configuration or created a broker domain in your Message Brokers Toolkit, you can use the server project that contains your broker domain configuration details, instead of creating a new server project.
- How can I create a broker archive?  
A broker archive file is a compressed file that contains compiled message flows, message dictionaries and other application development resources. Use this link to find out how to create a broker archive. You can use the “Broker Archive editor” on page 91 to add or remove message flows and message sets to or from your broker archive, and to edit the configurable properties within your bar file.
- How can I add resources to a broker archive?  
You can add message flows and message sets to a broker archive file. The message flows, message sets and related resources are compiled when they are

added to the broker archive file. You can also choose to add the source files to the broker archive file. You can not add resources to a broker archive file from a project that contains an error.

- How can I deploy a broker archive file?

Find out about the different ways that you can deploy a broker archive file to a broker. For deploying applications during development to a test environment, the simplest method is to use the Message Brokers Toolkit. When you deploy to a broker you select an execution group to deploy to.

- How do I check the results of deployment?

Use this link to find out how you can check the results of a deployment.

- How do I refresh the contents of a broker archive?

You can refresh the contents of a broker archive by removing resources from it and, having made required changes, add them back again. Alternatively you can use the Refresh option in the Broker Archive editor.

## How can I configure applications?

Use the links in this section to find out how to configure your applications for use in a production environment.

- How do I edit configurable properties?

System objects that are defined in message flows can have configurable properties that you can update within the broker archive file before deployment. By changing configurable properties, you can customize a broker archive file for a new domain, without needing to edit and rebuild the message flows or other application resources.

- How do I configure message flows at deployment time using user-defined properties?

User-defined properties (UDPs) give you the opportunity to configure message flows at deployment time, without modifying program code.

- How can I optimize message flow throughput?

Use this link to find out how you can optimize the throughput of messages in your message flows using parallel processing.

- How do I add multiple instances of a message flow to a broker archive?

You can edit the name of your files in the broker archive so that you can deploy multiple instances of a message flow with different values for the configurable properties.

- How do I configure global coordination of transactions?

To ensure data integrity during transactions, you can globally coordinate message flow transactions with a WebSphere MQ queue manager. Follow this link to find out how to configure your databases, queue manager and applications for global coordination.

## What are the other types of deployment?

Use the links in this section to find out how to deploy changes to your configuration or publish/subscribe settings to the broker.

- How do I deploy a broker configuration?

Find out how to deploy a broker configuration using the Message Brokers Toolkit, the mqsideploy command and the Configuration Manager Proxy API.

- How do I deploy a publish/subscribe topology?

Find out how to deploy a publish/subscribe configuration using the Message Brokers Toolkit, the mqsideploy command and the Configuration Manager Proxy API.

- How do I deploy a publish/subscribe topics hierarchy?

Find out how to deploy a publish/subscribe hierarchy using the Message Brokers Toolkit, the mqsideploy command and the Configuration Manager Proxy API.

## How do I administer and monitor a broker domain?

New users: use the links in this topic to get information about administering your broker domain.

### How do I administering my broker domain?

Use the links in this section to find out about the tasks that you operate frequently to activate and run your broker domain.

- How do I start and stop a broker?

Use this link to find out how to start and stop a broker on the different platforms. You can stop and start a broker using the command line.

- How do I start and stop the Configuration Manager?

Use this link to find out how to start and stop the Configuration Manager on the different platforms. You can stop and start the Configuration Manager using the command line.

- How do I start and stop the User Name Server?

Use this link to find out how to start and stop the User Name Server on the different platforms. You can stop and start the User Name Server using the command line.

- **How do I connect to a broker domain?**

Use the following links to find out how to connect to your broker domain:

- Using the Message Brokers Toolkit
- Using the Configuration Manager Proxy
- On z/OS

- **How do I start and stop message flows?**

Use the following links to find out how to start and stop message flows:

- Using the Message Brokers Toolkit
- Start and Stop message flows using the command line
- Using the Configuration Manager Proxy

- How do I deploy a broker archive file?

Use this link to find out how to deploy a broker archive file to a broker using the Message Brokers Toolkit, the mqsideploy command and the Configuration Manager Proxy API.

- How do I submit batch requests using the Configuration Manager Proxy?

You can use the Configuration Manager Proxy to group multiple requests destined for the same Configuration Manager together and submit them as a single unit of work.

- **How do I modify my components?**

You might want to change the configuration of your components in your broker domain. Use the following links to find out how to modify the components for your platforms.



- |                   – How do I modify a broker?
- |                   – How do I modify the Configuration Manager?
- |                   – How do I modify the User Name Server?
- |                   • **How do I delete components?**
- |                    Use the following links to find out how to delete components for your
- |                    platforms.
- |                    – How do I delete a broker?
- |                    – How do I delete the Configuration Manager?
- |                    – How do I delete a User Name Server?
- |                   • How do I back up my resources?
- |                    Follow this link to find out how to back up your WebSphere Message Broker
- |                    broker domain and application resources.

## | **How can I monitor my broker domain?**

| Use the links in this section to find out how you can monitor the events and  
| performance in your broker domain.

- |                   • How can I view broker domain log information?
- |                    The “Event Log editor” on page 96 in the Message Brokers Toolkit contains
- |                    information about events that occur within your broker domain. These events
- |                    can be information, errors, or warnings and relate to your own actions.
- |                   • How can I view message flow accounting and statistics data?
- |                    Message flow accounting and statistics data is the information that can be
- |                    collected by a broker to record performance and operating details of message
- |                    flow execution. Follow this link to get an overview of, and links to further
- |                    information, on message flow accounting and statistics.
- |                   • How can I generate statistics reports?
- |                    You can generate statistics reports that provide information about the
- |                    performance of your brokers. Use this link to find out about different types of
- |                    statistics reports and how to subscribe to them.
- |                   • What are the system management interfaces?
- |                    The brokers provide a service for independent system management agents so
- |                    that a central management facility can access information about any network
- |                    that includes a broker domain. Use this link to find out how you can use the
- |                    system management publications from WebSphere Message Broker to develop or
- |                    use system management adapters or customized administrative applications to
- |                    receive information on broker domain activity.

---

## | **How can I diagnose problems?**

| New users: use the links in this topic to find out how to debug message flow  
| applications and get help with troubleshooting problems.

### | **How can I debug and troubleshoot my message flow applications?**

| A number of tools and sources of information are available to help you solve  
| problems with your message flow applications. If you can see errors or warnings  
| in the Problems view, then your message flow applications and related resources  
| might have problems that prevent you from adding them to a broker archive file.  
| Also, your message flow applications might not work as you expect when they are  
| deployed. Use the information in the error or warning messages to fix the problem.  
| If you have difficulty deploying message flow application resources, see the

*Troubleshooting problems deploying message flow applications* section in this topic for more information. If you have successfully deployed your applications, you can use the following links to debugging and troubleshooting topics to help you solve any problems.

- How do I use the flow debugger?

You can test and debug your message flow applications using the debugger in the Message Brokers Toolkit. By adding breakpoints to your message flows, you can use the debugger to view the path that messages take through your message flow. You can also view the contents of the message passing through the flow, and step through code to view the output messages being constructed. The debugger is launched from the “Debug perspective” on page 87.

- Where can I find useful logs?

Find out where to find useful logs that are used in WebSphere Message Broker.

- How do I use trace?

User trace can be used to get further information about problems with messages passing through your message flows. You can use the following sequence of tasks to get user trace to troubleshoot a problem with a message flow:

1. Starting user trace
2. Retrieving user trace
3. Formatting trace
4. Interpreting trace

- How can I resolve problems when developing message flows?

Follow this link for information about some common problems when developing message flows.

- How can I resolve problems when developing message models?

Follow this link for information about some common problems when developing message models.

### **How can I troubleshoot problems deploying message flow applications?**

If your message flow applications or related resources contain an error, they cannot be added to a message broker archive file. Sometimes message flow applications successfully added to a broker archive file can fail to be deployed, for example, because of a problem with the broker or with the message flow application. Use the following links to find out how to check the results of deployment and get help solving problems.

- How can I check the results of deployment?

Find out how to check the results of a deployment using the “Event Log editor” on page 96, the `mqsdeploy` command, and the Configuration Manager Proxy API.

- How can I resolve problems when deploying message flows or message sets?

Follow this link to find tips on resolving problems when deploying message flows and message sets.

### **Where can I get other help on troubleshooting and solving problems?**

Use the links in this section to find out more information about troubleshooting and how to get support.

- What initial checks can I make?

Follow this link to get more in depth advice on checks to make if you have problems with WebSphere Message Broker.



- How do I deal with problems?  
Use the links in this topic to find out about common problems that you might see when you use WebSphere Message Broker. Check to see if any of these match the problems you are observing.
- Where can I get more help with problems?  
View this topic for links to more techniques for diagnosing problems with WebSphere Message Broker, and links to information about getting support.

---

## What information is available for users of previous versions?

If you are a user of a previous version of WebSphere Message Broker, use the links in this topic to get information about what is new in WebSphere Message Broker Version 6.0, and how to migrate from previous versions.

### What's new?

Use the links in this section to find out about the main enhancements and changes in this release. You can find out more about the new functions and capabilities of this release by taking a look at the new WebSphere Message Broker Version 6.0 samples in the Samples Gallery.

- “What’s new in Version 6.0?” on page 39  
Find out about the main new function in this release, including what is new in Version 6.0 fix packs, and new features added at Version 5.0.
- “How do I use the Message Brokers Toolkit?” on page 7  
Find out about the Message Brokers Toolkit if you are new to Eclipse or Rational Application Developer. Get information on navigating and customizing the Message Brokers Toolkit.
- Where can I find sample applications?  
Take a look at the Samples Gallery for new samples introduced in Version 6.0 demonstrating WebSphere Message Broker capabilities. Follow this link to get a description of all the samples and a direct link to each one in the Samples Gallery. You can view samples only when you use the information center that is integrated with the Message Brokers Toolkit.

### What do I need to know about migration and coexistence?

Use the links in this section to find out how to migrate to WebSphere Message Broker from previous versions of the product.

- How does WebSphere Message Broker Version 6.0 coexist with previous versions?  
WebSphere Message Broker Version 6.0 can coexist with previous versions. You can therefore perform a staged migration when you want to migrate from a previous version. See this topic for details on the components and versions that can coexist on the same machine.
- How do I migrate from Version 5.0 products?  
If you are migrating from Version 5.0 products see this topic for information on planning for migration, and the steps involved in carrying out a migration from your specific product.
- How do I migrate from Version 2.1 products?  
If you are migrating from Version 2.1 products see this topic for information on planning for migration, and the steps involved in carrying out a migration from your specific product.

- How do I upgrade from WebSphere Event Broker Version 6.0 to WebSphere Message Broker Version 6.0?

If you are upgrading from WebSphere Event Broker Version 6.0 see this topic for information on steps for performing the upgrade.

See the IBM Redbooks publication [Migration to WebSphere Message Broker v6](#) for more information about migrating to WebSphere Message Broker Version 6.0. This link works only if you have a connection to the Internet.

---

## Part 2. Product overview

<b>Product overview</b> . . . . .	37
WebSphere Message Brokers introduction . . . . .	37
Using WebSphere Message Brokers in your business . . . . .	38
What's new in Version 6.0? . . . . .	39
WebSphere Message Broker Version 6.0 . . . . .	57
Create the run time. . . . .	58
Develop applications . . . . .	59
Deploy applications to the run time . . . . .	59
Publish/subscribe . . . . .	60
Message Brokers Toolkit . . . . .	60
Runtime environment . . . . .	67
WebSphere Message Broker scenario . . . . .	74
Mergers and acquisitions scenario . . . . .	75
Accessibility features for WebSphere Message Broker	76



---

## Product overview

This section provides introductory information to help you get started with WebSphere Message Broker:

- “WebSphere Message Brokers introduction”
  - Quick Tour
  - “What’s new in Version 6.0?” on page 39
- “WebSphere Message Broker Version 6.0” on page 57
- “WebSphere Message Broker scenario” on page 74
- “Accessibility features for WebSphere Message Broker” on page 76
- Legal information

---

## WebSphere Message Brokers introduction

WebSphere Message Broker and WebSphere Event Broker belong to a family of business integration products that is available from IBM.

Business integration is the coordination and cooperation of all your business processes and applications. It involves bringing together the data and process intelligence in your enterprise, and harnessing these so that your applications and your users can achieve their business goals.

Business integration means that:

- You can connect customers, suppliers, partners, and service providers, with continuing security and control, to enable newly built and re-engineered applications for more effective business processes (for example, Supply Chain Management).
- You can make mergers and acquisitions a success by integrating dissimilar IT infrastructures from more than one company so that they can work together as a single entity.
- You can react more quickly to market trends and opportunities because your IT systems are flexible and dependable, and no longer constraining.
- You can overcome the barriers of diverse computer systems, geographic boundaries, time differences, language and format differences, and different methods of working.

WebSphere MQ messaging provides a secure and far-reaching communications infrastructure that you can expand with WebSphere Message Broker or WebSphere Event Broker to apply intelligence to your business data as it travels through your network.

### Transports

The main components of WebSphere Message Broker and WebSphere Event Broker (the broker, the Configuration Manager, the User Name Server, and the Message Brokers Toolkit) communicate using WebSphere MQ’s communications protocol, WebSphere MQ Enterprise Transport, or, if you implement WebSphere MQ Everywhere, WebSphere MQ Mobile Transport.

Your business applications, which can be on any of more than thirty industry platforms including those from IBM, Microsoft, and Sun Microsystems, Inc., can connect to the broker using WebSphere MQ protocols, or using other protocols,

such as WebSphere MQ Telemetry Transport, WebSphere MQ Real-time Transport, WebSphere MQ Multicast Transport, WebSphere MQ Web Services Transport, or WebSphere Broker JMS Transport.

The benefit of using WebSphere MQ protocols (WebSphere MQ Enterprise Transport or WebSphere MQ Mobile Transport) is that they provide assured, once-only delivery of messages between the components.

WebSphere MQ protocols provide rich support for applications:

- The Message Queue Interface (MQI) and Application Messaging Interface (AMI) are supported in several programming languages.
- The point-to-point (including request/reply and client/server) and publish/subscribe application communication models are supported.
- The complexities of communications programming are handled by the messaging services and are therefore removed from the application logic.
- The applications can access other systems and interfaces through adapters and gateways to products such as Lotus® Domino®, Microsoft Exchange/Outlook, SAP/R3, and CICS® and IMS/ESA® products.

## **WebSphere Event Broker**

WebSphere Event Broker provides an optimized publish and subscribe function within a limited WebSphere Message Broker framework. It provides high-performance nonpersistent publish and subscribe functionality to clients that are connected using various transports.

## **WebSphere Message Broker**

WebSphere Message Broker incorporates WebSphere Event Broker and extends its function to provide a powerful message broker solution driven by business rules. Messages are formed, routed, and transformed according to the rules defined by an easy-to-use graphical user interface (GUI).

Diverse applications can exchange information in dissimilar forms, with brokers handling the processing required for the information to arrive in the right place in the correct format, according to the rules that you have defined. The applications do not need to know anything except their own conventions and requirements.

Applications also have much greater flexibility in selecting which messages they want to receive, because they can specify a topic filter, or a content-based filter, or both, to control the messages that are made available to them.

WebSphere Message Broker provides a framework that supports supplied, basic, functions along with user-defined enhancements, to enable rapid construction and modification of business processing rules that are applied to messages in the system.

## **Using WebSphere Message Brokers in your business**

WebSphere Event Broker addresses the needs of business and application integration by distributing real-time information from disparate sources of information through a network of access points or a centralized broker. It allows you to:

- Publish a message to make it available to other applications. Other applications can choose to receive publications that relate to specific topics, or that have specific content, or both.
- Create structured topic names, topic-based access control functions, content-based subscriptions, and subscription points.

WebSphere Message Broker addresses the needs of business and application integration by managing the flow of information. In addition to the function of WebSphere Event Broker, it provides services, based on message brokers, to allow you to:

- Route a message to several destinations, using rules that act on the contents of one or more of the fields in the message or message header.
- Transform a message, so that applications using different formats can exchange messages in their own formats.
- Store a message, or part of a message, in a database.
- Retrieve a message, or part of a message, from a database.
- Modify the contents of a message; for example, by adding data extracted from a database.
- Exploit a public interface to develop message processing node types that can be incorporated into the broker framework to complement or replace the supplied nodes, or to incorporate node types developed by Independent Software Vendors (ISVs).
- Enable instrumentation by products such as those developed by Tivoli, using system management hooks.

The benefits of WebSphere Message Broker can be realized both within and outside your enterprise:

- Your processes and applications can be integrated by providing message and data transformations in a single place, the broker. This helps reduce the cost of application upgrades and modifications.
- You can extend your systems to reach your suppliers and customers, by meeting their interface requirements within your brokers. This can help you improve the quality of your interactions, and allow you to respond more quickly to changing or additional requirements.

## Further information

For more information about WebSphere business integration, see the WebSphere Business Integration Web page. For a basic introduction to WebSphere Message Brokers, see the IBM Redbooks publication WebSphere Message Broker Basics.

## What's new in Version 6.0?

This topic introduces you to the main new function in WebSphere Message Broker Version 6.0. See also "New function added in Version 6.0 fix packs" on page 49.

- Extended platform support
- Simplified installation and migration
- Improved Message Brokers Toolkit
- Flexible transformation
- Improved performance and scalability
- Coexistence
- Enhanced support for Web services

- Driving message flows
- Enhanced message parsing
- 
- Improved management
- Improved security
- License management
- Code page conversion
- WebSphere MQ as a transaction manager

#### **Extended platform support**

The Configuration Manager is supported on all the broker platforms including z/OS, UNIX, Linux, and Windows; see “Configuration Manager” on page 68.

The Message Brokers Toolkit can be installed on Linux (x86 platform) in addition to Windows; see “Message Brokers Toolkit” on page 60.

Database support on Linux (x86 platform) is extended to include Oracle in addition to UDB DB2; see Supported databases.

#### **Simplified installation and migration**

The installation of WebSphere Message Broker Version 6.0 is easier than that of WebSphere Business Integration Message Broker Version 5.0 because the number of prerequisite products has been reduced. For more information on installation, see the Installation Guide.

Configuration Managers no longer require a database.

A Default Configuration wizard is provided to quickly build a full environment for development or other simple configurations on Windows and Linux (x86 platform). For more information on the Default Configuration wizard, see Using the Default Configuration wizard, and the Installation Guide.

Migration from WebSphere MQ Integrator Version 2.1 and WebSphere Business Integration Message Broker Version 5.0 are supported. Both Version 2.1 and Version 5.0 can coexist with WebSphere Message Broker Version 6.0, allowing a phased migration. In addition, the command `mqsimigratecomponents` is provided to migrate individual components from one codebase to another. For more information on migration, see Migrating and upgrading.

The installation of WebSphere Message Broker for z/OS Version 6.0 has been simplified in the following ways:

- A single command can be used to create brokers, Configuration Managers, and User Name Servers.
- All commands are JCL-based, meaning that no UNIX shell is required.

For more information, see the Installation Guide.

#### **Improved Message Brokers Toolkit**

The Message Brokers Toolkit is based on the latest release of Rational Application Developer, powered by the Eclipse open source platform. The Message Brokers Toolkit includes the following new features:

- Support on Linux (x86 platform) in addition to Windows.
- XML editors.
- Visual debugging with the following resources, which reduces the need for Eclipse perspective switching:



- Message flows
- ESQL
- Java
- Graphical mappings

For more information, see Flow debugger overview.

For more information about the Message Brokers Toolkit, see “Message Brokers Toolkit” on page 60.

## Flexible transformation

### Enhanced graphical mapping

The mapping tools support both novice and expert users, when authoring unidirectional transformations between source and target instance data elements. In addition to requiring less ESQL coding, the tools include the following enhancements:

- A Message Mapping editor with a source pane, target pane, drag and drop, spreadsheet view, expression entry field, edit window, and marker bar with breakpoint indication for debug purposes; see “Message Mapping editor” on page 107.
- An incremental builder that validates map content and external links, and generates ESQL object code as output for deployment to the broker.
- Support for user-defined functions written in Java and ESQL.
- The ability to split source messages into a number of output messages, convert the message type automatically, map message headers, and map repeating elements.

Compatibility with earlier versions is provided by automatic migration, where existing mfmap files are rewritten in the new msgmap format. Message flows with Mapping nodes continue to function without change.

For more information about mapping, see Message mappings overview.

### Routing and transformation rules in Java

You can use the JavaCompute node to write routing and transformation logic in Java. A standard J2SE 1.4.2 environment is provided with the standard Eclipse Java editor with color highlighting and code assist to create the transformations. XPath helper methods are provided, in addition to the full plug-in methods, to give easy access to message fields. No ESQL skill or experience is required.

You can deploy Java jars to the broker using the standard deployment facilities.

For more information about the JavaCompute node, see JavaCompute node.

### ESQL enhancements

Transforming and routing messages using ESQL has been made easier by the following enhancements:

- In-memory cache to reduce access to databases for read-only routing or validation data.
- Improved support for creating DATETIME variables.

- Access to multiple databases from the same Compute, JavaCompute, Database, or Filter node.
- Dynamic database schemas.
- New-user defined properties that can pass parameters to ESQL to modify standard behavior.
- Access to environment information; for example, message flow name and broker name.
- Improved support for result sets returned by database stored procedures.
- Improved error recovery using SQL handlers.
- Multiple out terminals to combine the function of Compute and Filter nodes.

For more information, see ESQL overview.

#### **XSLT enhancements**

Performance of XSLT transformations is improved by the support of compiled style sheets. The style sheets can also be deployed using the standard deployment facilities.

For more information on XSLT transformations, see XMLTransformation node.

#### **Improved performance and scalability**

Performance of the broker run time has been significantly improved by the following enhancements:

- The path lengths of the major broker functions have been shortened.
- The cost of parsing and streaming messages has been reduced.
- ESQL and publish/subscribe functions have been improved.
- The aggregation nodes now use WebSphere MQ queues to store state information instead of a database. This improves the throughput of all requests, with the greatest improvement being gained with non-persistent requests.
- A new in-memory cache allows more efficient message flows to be developed.
- The storage requirements of the MRM and XML parsers have been reduced so that larger messages can be processed.

#### **Coexistence**

WebSphere Message Broker Version 6.0 can coexist with either a Version 2.1 or a Version 5.0 product on the same computer. You can install WebSphere Message Broker Version 6.0 in a different location on the same computer, migrate your components and resources to WebSphere Message Broker Version 6.0, and uninstall the Version 2.1 or Version 5.0 product later when you are sure that you no longer need it.

With some restrictions, all Version 5.0 components can participate in a Version 6.0 broker domain, and all Version 6.0 components can participate in a Version 5.0 broker domain. A Version 2.1 broker is the only Version 2.1 component that can take part in a Version 6.0 broker domain.

For more information about coexistence, see Coexistence with previous versions and other products.

#### **Enhanced support for Web services**

Web services support has been extended in the following ways:

- SOAP 1.2 is supported.

- HTTP 1.1 is supported.
- SOAP schemas are provided; using SOAP schemas can help simplify the modeling and transformation of SOAP messages; see Web services applications - SOAP.
- WSDL definitions can be imported using the new WSDL importer; see Importing from WSDL files to create message definitions.
- The WSDL generator has been improved; see Generating a Web Service Definition from a message set.
- Imported and generated WSDL is validated for compliance with the Web Services Interoperability (WS-I) Basic Profile 1.0.
- The HTTP transport has been extended to provide HTTPS support, which provides added privacy and security.

### **Driving message flows**

Message flows can be driven by JMS transports. Supported JMS providers include the embedded JMS provider in WebSphere Application Server Version 6.0; see WebSphere Broker JMS Transport.

New TimeoutControl and TimeoutNotification nodes allow message flows to be driven periodically rather than by an external event; see TimeoutControl node and TimeoutNotification node.

A new MQGET node allows messages to be retrieved midway through a message flow, not only at the beginning of the message flow. Groups of messages can be dealt with together, and queues can be used to save temporary state information; see MQGet node.

### **Enhanced message parsing**

The performance of both parsing and writing messages has been significantly improved to allow greater throughput, especially of large messages.

A new WSDL importer is provided to create message models from WSDL files, simplifying the integration of Web Services.

A new MIME parser is provided to allow the parsing of multipart MIME messages such as SOAP with Attachments and RosettaNet; see MIME parser and domain.

Message models for industry standard message definitions, such as SOAP envelope, MIME headers, and SAP IDoc segments, are supplied for inclusion in your own message sets.

Support has been added for XML Schema list and union simple types, xsi:type attributes, and XML version 1.1.

Unbounded repeats are supported for all kinds of message including binary and formatted text.

Enhanced support is provided for COBOL messages that use OCCURS DEPENDING ON.

The following runtime validation improvements are provided:

- Validation options on more nodes.
- A new Validate node to validate a message in the middle of a message flow.
- The ability to detect all validation failures in a message, before throwing an exception.

- The ability to force a complete parse of a message independently of runtime validation.

## **Improved management**

### **Runtime versioning**

The enhanced versioning capabilities that were introduced to the development environment in the last release have been extended to the runtime environment. All deployed resources can be tagged with version, author, and other useful information in addition to the standard compiled time and deployment time attributes. The new information is displayed in the administration interface, making it easy to see which resources have been deployed. For more information on runtime versioning, see *Message flow version and keywords*.

### **More automation**

New and extended command line utilities are provided to allow full automation of the deployment to production environments of new resources.

The Configuration Manager Proxy (CMP) Java API is a new systems management application interface that is provided to allow WebSphere Message Broker to be fully managed by products and utilities other than the Message Brokers Toolkit and the command-line interface. For more information on the CMP, see *Developing applications that use the Configuration Manager Proxy Java API*.

## **Improved security**

The HTTP transport has been extended to provide HTTPS support, which provides added privacy and security.

Communication between the Message Brokers Toolkit and a Configuration Manager has been extended to allow the use of SSL for added security.

## **License management**

WebSphere Message Broker supports a new form of license management using IBM Tivoli License Manager (ITLM), Version 2.1, across all the WebSphere Message Broker platforms except z/OS. Implementing ITLM allows sub-capacity pricing for eBusiness On Demand.

For more information on ITLM, see *Installing Tivoli License Manager*.

## **Code page conversion**

The set of code page converters that are used by WebSphere Message Broker has been updated and is now comprised of code page converters from the International Components for Unicode (ICU) libraries for Unicode, Version 3.2. For the list of code page converters that are included in the set, see *Supported code pages*.

For information on how to add additional code page converters to the set, see *Generating a new code page converter*.

## **WebSphere MQ as a transaction manager**

When using WebSphere MQ Version 6.0 as a transaction manager, data sources in coordinated message flows cannot connect to 32-bit DB2 instances. If data sources in your coordinated message flows connect to DB2, ensure that they only connect to 64-bit DB2 instances.

## **New function added in Message Brokers Toolkit Version 6.0.2.0**

While the “What’s new in Version 6.0?” on page 39 topic introduces you to the main new function in WebSphere Message Broker Version 6.0, “New function added in Version 6.0 fix packs” on page 49 summarizes new function in its fix packs, and this topic introduces you to additional function that has been added in Message Brokers Toolkit Version 6.0.2.0. This version is compatible with any runtime component level.

- Start here topics in the information center
- New Quick Start wizards
- New Message Set wizard
- New Message Flow wizard
- New XML Schema wizard renamed
- RDB Definition Files wizard renamed
- New Message Definition wizard redesigned
- New WSDL Definition wizard renamed
- Option to generate XML Schemas and WSDL to an external folder
- Creation of a message definition file based on a C header, a COBOL copybook , or an XML DTD file
- Support for mapping MIME messages
- Broker Development view (formerly the Resource Navigator view)
- Displaying projects in working sets
- Global toolbar in the Broker Application Development perspective
- Add a node by dragging from the Broker Development view (formerly Resource Navigator view)
- New categories for nodes on the node palette
- Editing node properties
- Mapping enhancements
- Refreshing the contents of a broker archive
- Test Client

### **Start here topics in the information center**

A Start here category has been added to the Contents pane of the information center. The Start here category contains topics that provide an introduction for users who are new to WebSphere Message Broker Version 6.0. Use the Start here topics to get overview information and links to relevant topics in the information center on getting started, developing applications, broker administration, and troubleshooting tasks.

### **New Quick Start wizards**

You can use the three new “quick start” wizards to help you to create your applications using the Message Brokers Toolkit. For more details, see Getting started with Quick Start wizards.

### **New Message Set wizard redesigned**

The New Message Set wizard has been redesigned, therefore you do not need to create a message set project when you create a new message set. A more logical tree structure is used by the Message Set editor to organize the physical formats of message sets. See Creating a message set for more details about the New Message Set wizard.

### **New Message Flow wizard redesigned**

The New Message Flow wizard has been redesigned to allow you to either

select an existing message flow project for a new message flow, or to create a new message flow project for the new message flow. See [Creating a message flow](#) for more information.

#### **New XML Schema wizard renamed**

The New XML Schema wizard is now called the Generate XML Schema wizard. To start the wizard, click **Generate** → **XML Schema**. For more information, see [Generating an XML schema](#).

#### **RDB Definition Files wizard renamed**

The RDB Definition Files wizard is now called the New Database Definition Files wizard. To start the wizard, click **New** → **Database Definition Files**.

#### **New Message Definition File wizard redesigned**

The New Message Definition File wizard has been redesigned. Click **New** → **Message Definition File** to create a new message definition file from scratch, or click **New** → **Message Definition File From** to create a new message definition file from existing resources. In the latter case, you then choose from a list of existing resources (XML Schema file, IBM-supplied message, XML DTD file, C header file, COBOL file, or WSDL file). See [Creating a message definition file from scratch](#) and [Creating a message definition file from an existing resource](#) for more details about the new message definition wizards.

When you create a new message definition file from an existing resource such as an XML Schema file, you can now select a file from outside the current workspace (previously, the file had to exist in the current workspace).

The steps to create a new message definition file from a WSDL file have been simplified.

#### **New WSDL Definition wizard renamed**

The New WSDL Definition wizard is now called the Generate WSDL Definition wizard. To launch the wizard, click **Generate** → **WSDL Definition**. For more details, see [Generate WSDL Definition wizard](#).

#### **Option to generate XML Schemas and WSDL to an external folder**

You can now choose an external folder as the destination for generated XML Schemas and WSDL definitions.

#### **Creation of a message definition file based on a C header, a COBOL copybook , or an XML DTD file**

The wizards that you use to create a message definition file that is based on a C header file, a COBOL copybook, or an XML DTD file, have all been simplified.

#### **Support for mapping MIME messages**

You can create message mappings for messages in the MIME message domain. Select the MIME message domain for your message set, and create the message mappings that you require. If you plan to create message mappings for messages in the MIME domain, your messages must comply with the rules stated in the following topics:

- MIME parser and domain
- MIME standard header fields
- MIME messages

#### **Broker Development view (formerly the Resource Navigator view)**

The Resource Navigator view in the Broker Application Development



perspective has been renamed and is now called the Broker Development view. The ways in which resources can be displayed in the Broker Development view have also changed. For example, you can choose whether to display or hide namespaces, categories, or categorized file extensions. When you create resources, they are grouped together by file type in logical categories, and placed in virtual folders. You can choose whether to display the resources in the logical categories, or to hide them. You can also filter closed projects in the View menu. See “Resources” on page 63 and “Broker Application Development perspective” on page 83 for more information.

### **Displaying projects in working sets**

A working set is a logical collection of application projects, which you can use to limit the number of resources that are displayed in the Broker Development view. For more information, see “Resources” on page 63 and Displaying selected projects in working sets.

### **Global toolbar in the Broker Application Development perspective**

The icons that are displayed by default in the toolbar of the Broker Application Development perspective have changed. See “Broker Application Development perspective” on page 83 for more information.

### **Add a node by dragging from the Broker Development view (formerly Resource Navigator view)**

You can add a message flow node to the canvas by dragging it from the Broker Development view, as described in Dragging a resource from the Broker Development view.

### **New categories for nodes on the node palette**

The nodes on the node palette have been organized into categories, as shown in Message flow node palette. You can add the nodes that you use most often into the Favorites category, either by dragging them, or using the menu, as described in Adding nodes to the Favorites category on the palette.

### **Editing node properties**

- You can rename a node in the Message Flow editor when you add it to the canvas, or you can rename it on the Description tab of the Properties view.
- When you double-click a node in the Message Flow editor, you open the resource that is associated with that node; for example, if you double-click a Mapping node, you open the New Message Map dialog box. When you double-click a node that does not have a resource associated with it, you open the Properties view for that node.
- You can choose to view and edit a node’s properties either in the tabbed Properties view beneath the Message Flow editor, or in the Properties dialog box.

For more details, see Configuring a message flow node.

### **Mapping enhancements**

- The New Message Map wizard has been changed. Select the type of message map that you want to create, then select the sources and targets for the message map. The options available in the New Message Map wizard depend on the node that you have selected, or whether you are creating a message map from the Broker Development view (formerly Resource Navigator view). For more details see the following topics:
  - Creating a message map file in the Broker Development view

- Creating a message map file from a Mapping node
- Creating a message map file from a DataInsert node
- Creating a message map file from a DataUpdate node
- Creating a message map file from a DataDelete node
- You can no longer select to map headers in your message map from the New Message Map wizard. Instead, you can now select to add the following message headers and folders to your message map using the Add or Remove Headers and Folders dialog box:
  - LocalEnvironment
  - Properties
  - MQ Headers
  - HTTP Headers
  - JMS Transport Header

Properties are included with the message body in a message map by default. If you do not select to add the LocalEnvironment or any extra message headers to your message map, the corresponding values are copied from the source message to the target message automatically. For more details see Mapping headers and folders. You can also convert between a message map and a submap using the Add or Remove Headers and Folders dialog box as described in Converting a message map to a submap.

- Enhancements have been made to mappings involving databases:
  - When you create a new message map that has a database table as a target, you must select the database table from the list of database operations.
  - You can now change the database operation of a message map that modifies a database as described in Change database operation of a message map.
  - You can now create a database submap to modify a database as described in Creating a submap to modify a database.
  - You can now create a database-to-database mapping, that uses a source database to modify a target database table, as described in Creating a database to database mapping.
- Two new dialog boxes have been added to the “Message Mapping editor” on page 107. These dialog boxes, in addition to the Add Sources and Targets dialog box, can be used to add sources and targets to your message map. You can use the Add Sources dialog box to add Messages, Message Components, and Data Sources to your message map as sources. You can use the Add Targets dialog box to add Messages, Message Components, and Data Targets to your message map as targets. See the following topics for more information:
  - Adding messages or message components to the source or target
  - Adding a database as a source or target
- You can drag sources and targets from the Broker Development view (formerly Resource Navigator view) onto the Source and Target panes in your message map. See the following topics for more information:
  - Adding messages or message components to the source or target
  - Adding a database as a source or target

### **Refreshing the contents of a broker archive**

You can now refresh the contents of a broker archive in the Broker Archive



editor. This single task combines the actions of removing, updating, and replacing resources that were previously required. You can also now view, and clear, the user and service logs from the Broker Archive editor.

### **Test Client**

Test messages during message flow development using the Test Client in the Message Brokers Toolkit. Click **Test Message Flow** or **Test Client** on the menu that is displayed when you right-click a message flow file in the Broker Development view, or click **Test** when you right-click either an MQInput node or an HTTPInput node in the Message Flow editor to generate the Test Client editor.

You can use the Test Client editor to:

- Configure a Message Broker test environment
- Compose and edit test messages
- Start and send test messages
- Monitor test execution events
- Save the test configurations for reuse and analysis
- Save and review message test execution traces

For more details, see [Testing message flows using the Test Client](#)

### **New function added in Version 6.0 fix packs**

Whereas the “What’s new in Version 6.0?” on page 39 topic introduces you to the main new function in WebSphere Message Broker Version 6.0, this topic introduces you to the additional function that has been added in fix packs.

For detailed information about the content of fix packs, go to the WebSphere Message Broker support Web page and select your product to display the recommended fix pack, or to list all available fix packs. Each fix pack includes a memo, called memo.ptf, that includes details of its content.

### **Fix Pack V6.0.0.3 enhancements**

#### **WebSphere Message Broker Extensions, Samples, and Demos**

A set of optional product extensions is provided with Fix Pack V6.0.0.3 on a supplemental CD. You can install from the CD any of the following options:

##### **Extensions**

Choose from a selection of SupportPacs that extend and enhance broker support for your applications. These extensions include additional message flow nodes and utilities that interact with the broker.

##### **Samples**

Additional samples demonstrate specific broker functions and complement the samples that are in the Samples Gallery.

##### **Demos**

Demos give a visual explanation of broker features to help you to understand how you can create solutions to your business challenges.

The CD provides a graphical interface that describes its content, provides information about the resources that helps you to make decisions about what you want to install, and completes the installation of those resources that you want.

When you install a resource, the CD also installs the documentation that is associated with that resource and describes its use. For some resources, additional manual installation and setup instructions are provided, and you must complete these steps to take advantage of the installed resource.

The CD also includes a number of links to other sources of information, for example Web sites, that are available to help you to use the function of WebSphere Message Broker.

If you are registered with Passport Advantage, you can also download an electronic image of the WebSphere Message Broker Extensions, Samples, and Demos CD.

#### **Java 5 SR2 support**

Support is added for Java 5 (also known as Java 1.5) SR2.

- On distributed systems, both Java 1.4.2 and Java 5 are installed with WebSphere Message Broker. Java 1.4.2 is configured as the default version; if you prefer to use Java 5, you can modify your system, as described in Setting up a command environment.
- On z/OS, you must obtain and install Java 1.4.2, Java 5, or both. When you customize your component definitions, specify which Java installation is accessed by each component; see Customizing the level of Java.

#### **Specify SVRCONN channel on Configuration Manager's queue manager**

Support is added for naming the server-connection channel on the Configuration Manager's queue manager. This option enables you to set up more than one channel, each having a different level of security for users. See the following topics:

- Creating a domain connection
- Modifying domain connection properties
- Enabling SSL on the WebSphere MQ Java Client

#### **WebSphere Message Broker plug-in now available for the IBM Support Assistant (ISA)**

A WebSphere Message Broker plug-in is now available for ISA, which helps you to resolve questions and problems with IBM software products by providing access to support-related information and troubleshooting tools. See:

- IBM Support Assistant (ISA)
- Installing the IBM Support Assistant (ISA)
- Updating the IBM Support Assistant

#### **Support for 2007 Daylight Savings Time changes in runtime components**

Because of changes to the start and end of Daylight Savings Time (DST) in the U.S. in 2007, an environment variable MQSI\_USE\_NEW\_US\_DST is provided. See Observing 2007 U.S. changes to Daylight Savings Time.

#### **Fix Pack V6.0.0.2 enhancements:**

##### **Extended database support**

Support is added for the following databases on Solaris (x86 platform):

- Oracle 10G
- Sybase 12.5.3

See Supported databases for details of available database support by operating environment.

See Configuring databases to see that tasks involved in database configuration.

### **Mapping enhancements**

- You can select WebSphere MQ constants as target element values. See Setting the value of a target element to a WebSphere MQ constant.
- Several changes have been made to the wizards in the Mapping node:
  - The new Map by Name wizard performs automatic mapping of sources and targets that have the same or similar names. See Mapping from source: by name.
  - The Add Instance and Add Group Instances wizards have been replaced by the Insert Before and Insert After wizards that allow mapping statements to be added at specific locations in a map.
  - The Substitute Element wizard has been replaced with the Replace wizard that allows mapping statements on elements, attributes, and database columns to be substituted by similar statements.

### **IPv6 support**

Internet Protocol Version 6 is now supported. You can enter, or refer to, an IPv6 address in all situations in which an IPv4 address is valid; for example, in the mqsischangeproperties command.

### **Overview of creating z/OS components**

The information that describes the mandatory and optional tasks that create components on z/OS has been enhanced. No new function is provided, but the topics have been changed and extended to improve the order and description of how you achieve the results that you want in your broker domain. See Creating WebSphere Message Broker components on z/OS.

### **Fix Pack V6.0.0.1 enhancements:**

#### **Updated Rational support for Message Brokers Toolkit**

Message Brokers Toolkit is now based on Rational Application Developer (RAD) Version 6.0.1.1 and is compatible with that release and later Version 6.0.1 updates. It cannot coexist with RAD products that are based on different versions, so such products cannot be below Version 6.0.1.1, and must be below Version 6.0.2. Other Rational products must also be at the latest available level. For detailed guidance about ensuring that you have a compatible Rational framework, see the Installation Guide.

#### **Support for 64-bit execution groups**

The addition of support for 64-bit execution groups allows the use of very large messages, and also enables WebSphere Message Broker Version 6.0 to run applications in fastpath (trusted) mode when using the 64-bit queue manager that is provided in WebSphere MQ Version 6.0.

#### **Extended platform and operating system support for runtime environments**

Support is now provided for the following runtime environments:

- Solaris (x86-64 platform), running Solaris 10.
- Linux (POWER platform) on iSeries and pSeries hardware, running either of the following operating systems:
  - Linux PowerPC Red Hat Enterprise Advanced Server V4
  - Linux PowerPC SUSE Linux Enterprise Server (SLES) 9
- Linux Red Hat Enterprise Advanced Server V4 on Linux (x86 platform) and Linux (zSeries platform).
- Solaris 10 on Solaris (SPARC platform) and Solaris (x86-64 platform).

See Operating system requirements for more detailed information about supported environments.

### **Extended database support**

Support is added for the following databases:

- Remote access to Microsoft SQL Server 2000, from the following platforms:
  - AIX
  - HP-UX
  - Linux (x86 platform)
  - Solaris (SPARC platform)
- Informix Dynamic Server V9.4, on the following platforms:
  - AIX
  - HP-UX
  - Linux (x86 platform)
  - Solaris (SPARC platform)
  - Windows.
- IBM DB2 V8 on Solaris (x86-64 platform) and Linux (POWER platform).

See Supported databases for details of available database support by operating environment.

### **Backward recovery is extended to Version 2.1**

You can restore components and resources that you have migrated to Version 6.0 from Version 2.1 products back to their original state. For details of how to do this, see Restoring components and resources to Version 2.1

### **User exit for tracing message flows**

Support is provided for user exits that enable user-provided custom software to track data that passes through message flows in WebSphere Message Brokers. The user-provided functions can be invoked at specific points during the lifecycle of a message as the message passes through the message flow, and can invoke utility functions to query information about where the message is in the flow, and the contents of the message assembly. For more information about user exits see User exits

## **New in Version 5.0**

This topic introduces you to the main new function in WebSphere Business Integration Message Broker Version 5.0. See also “New function added in Version 5.0 fix packs” on page 55.

- New graphical development environment based on Eclipse

- Enhanced initial user experience
- Web Services support
- Enhanced message modeling
- XSLT transformation enhancements
- Extended database user ID and password support
- Message flow accounting and statistics
- Publish/subscribe enhancements
- SSL authentication, QoS, and HTTP tunneling
- New object level security model
- Increased platform support

### **New graphical development environment based on Eclipse**

The Message Brokers Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform. The Message Brokers Toolkit consists of the Eclipse platform and a set of Java plug-ins that enable the creation, maintenance, and deployment of WebSphere Message Broker message flows (including publish/subscribe applications) and message models. See the “Message Brokers Toolkit” on page 60 topic, and the Eclipse web site.

The Message Brokers Toolkit includes, among other things, the following new features:

- You can control multiple broker domains using a single Message Brokers Toolkit. See “Message Brokers Toolkit” on page 60.
- You can use Message Brokers Toolkit with external source repositories, for example with repositories that permit version control of message flow and message set source data. See “Development repository” on page 66.
- Improved editors, such as Message Flow Mapping editor, Message Set editor, and ESQL editor, make it easier for you to build your solution. See “Editors in the Message Brokers Toolkit” on page 90.
- The Mapping node allows you to create message transformations using a drag-and-drop method, without the need to write ESQL code. See Mapping node.
- You can now reuse ESQL subroutines and functions. See Broker schemas.
- The flow debugger allows you to add breakpoints to the connections of a message flow, and step through ESQL code statement by statement. See Flow debugger overview.
- You can configure some message flow properties at deployment time rather than at development time. See Configurable properties of a broker archive .

### **Enhanced initial user experience**

The **pre-install LaunchPad** is launched from the Windows CD prior to installation. It guides you through the process of checking for, and installing, any prerequisite software before launching the product installation. The installation is now implemented using InstallShield for Multi-Platforms.

The **Welcome** page for the Message Brokers Toolkit, which appears after you launch the Message Brokers Toolkit, provides links to a number of basic tasks, enabling you to configure and exercise a working system as quickly as possible. Supported tasks include:

- Take the Quick Tour.
- Create the Default Configuration.
- Verify your installation with the Pager samples.
- Explore the Samples Gallery.

Links to the Default Configuration wizard, samples, and Samples Gallery work only if you are accessing the WebSphere Message Broker documentation from the Message Brokers Toolkit.

### **Web Services support**

Several enhancements provide support for Web Services:

- HTTPInput, HTTPReply, and HTTPRequest nodes. For more information, see Built-in nodes.
- Web Service Description Language (WSDL) generation from a message set. For more information, see Generate WSDL.
- Full support for the construction and parsing of SOAP messages, and the addition of XML namespace support through the extension of the MRM XML Wire Format and the addition of the XMLNS domain. For more information, see WSDL generation, Namespaces in the message model, and The XML domains.
- See also WebSphere Business Integration Message Broker Version 5.0 and Web Services SupportPac (IA81).

### **Enhanced message modeling**

The following enhancements have been made to the message model:

- New logical message model based on the industry standard XML Schema 1.0. See The message model.
- Introduction of message definition files to make it easier to manage large message sets. See Message definition files.
- Full support for XML namespaces in the MRM domain and in the new XMLNS domain. See Namespaces in the message model and The XML domains.
- Introduction of user-defined simple types. See Message model objects: simple types.
- New XML Schema, XML DTD, C, and COBOL importers. See Importing data structures.
- Introduction of an XML Schema generator. See Generate XML Schema.

### **XSLT transformation enhancements**

The XML Transformation node allows you to integrate existing XSLT transformations into the broker environment. The style sheet used to define the transformation can be specified as an attribute of the node instance, or can be determined within the context of the flow and specified using a defined area of the Local Environment. This capability is delivered through repackaging and enhancement of the XALAN-based XSLT style sheet transformation engine derived from the WebSphere Transcoding Publisher product.

For more information, see XMLTransformation node.

### **Extended database user ID and password support**

External database access from Compute, Database and Warehouse nodes has been extended so that you can associate a specific user ID and password with a given ODBC DSN in the broker run time. This user ID

and password combination is then used for all connections made by the broker to the particular DSN, overriding the default use of the broker's user ID and password.

This addresses a key requirement to be able to specify different user ID and password combinations for different databases and to be able to set this on a broker-by-broker basis. The `mqsisetdbparms` command is provided for setting this information in the broker.

For more information, see [Accessing databases from message flows](#).

### **Message flow accounting and statistics**

You can now collect statistics on the behavior of message flows. For example, you can collect data about how many messages are processed and how large those messages are, or about CPU usage and elapsed processing times. For more information, see [Message flow accounting and statistics data](#), and the [developerWorks](#) article on [message flow performance](#).

### **Publish/subscribe enhancements**

The following publish/subscribe changes have been made:

- The set of protocols and distribution patterns supported by the broker has been extended to provide reliable, real-time IP multicast distribution of subscriptions over a Local Area Network. This is provided as an extension to the existing JMS IP support in WebSphere Event Broker. For more information, see [WebSphere MQ Multicast Transport and Multicast publish/subscribe](#), and [WebSphere MQ Real-time Transport](#).
- Publish/subscribe capabilities have been consolidated within WebSphere Message Brokers. This rationalizes the product offerings and provides a clear upgrade path from WebSphere Event Broker to WebSphere Message Broker.

### **SSL authentication, QoS, and HTTP tunneling**

The following Internet-related items have been added:

- SSL authentication provides additional security. See [SSL authentication](#).
- WebSphere MQ Telemetry Transport offers a range of message delivery options. See [WebSphere MQ Telemetry Transport Quality of Service levels and flows](#).
- HTTP tunneling allows applet access through firewalls. See [Tunneling](#).

### **New object level security model**

You can now control access to runtime resources by object as opposed to by group. See [Authorization to access runtime resources](#).

### **Increased platform support**

Supported operating environments now include Linux (x86 platform), Linux (zSeries platform), and Windows XP, giving you a wider choice of platforms on which to deploy your solution. For more information, see [Operating system requirements](#).

**New function added in Version 5.0 fix packs:** While the "New in Version 5.0" on page 52 topic introduces you to the main new function in WebSphere Business Integration Message Broker Version 5.0, this topic introduces you to some of the other new function added in fix packs.

For detailed information about the content of fix packs, go to the [WebSphere Message Broker support Web page](#) and select your product to display the



recommended fix pack or to list all available fix packs. Each fix pack includes a memo, called `memo.ptf`, that includes details of its content.

Some features added in Fix Pack 8:

- The aggregation nodes can now be configured to use WebSphere MQ queues for storing aggregation requests, instead of using the broker database. This improves the throughput of all requests, yet the external functioning of aggregation nodes remains unchanged. Using WebSphere MQ also allows aggregation to be run in a non-persistent mode when persistence of aggregation requests is not required. For more information about aggregation, see [Message flow aggregation](#).
- Enhancements to database support:
  - Oracle 10G support is added on AIX, HP-UX, Solaris, and Windows.
  - Oracle 9i and Oracle 10G support is added on Linux on Intel.
  - DataDirect Version 5.0 ODBC drivers are supported. These are required if you are using WebSphere MQ Version 6 as an XA Transaction Manager.
  - Connection can be made to 64-bit database clients for DB2 (Version 8.1 or later). This requires that you are using WebSphere MQ Version 6 on the broker system, and the DataDirect Version 5.0 ODBC drivers.

For detailed information about the database support available in WebSphere Business Integration Message Broker Version 5.0, see [Supported databases](#)

Some features added in Fix Pack 5:

- On Linux (x86 platform), you now use the DataDirect Driver Manager to connect to DB2 databases. For more information, see [Connecting to a database from Linux and UNIX systems](#).
- The MRM domain TDS physical format can now recognize and interpret HL7 service strings, and a new TDS messaging standard called 'HL7' has been added. For more information, see [HL7 messaging standard](#).
- The broker now supports the 'Z' character to mean a zero time difference from Coordinated Universal Time (UTC) when it appears as a time zone in a `dateTime` field in a message. It is possible to accept `dateTimes` containing 'Z' on input, and to generate them on output. For more information, see [DateTime as string data](#).
- The new `mqsigratequeues` command makes it easier to migrate a Version 2.1 broker to Version 5.0 by creating any new queues that are needed by the updated broker.
- The `mqsigratemsgsets` command includes a new option `-pl` that you can use to specify that elements that have prefixed identifiers and are referenced more than once are to be created as local rather than global. For more information, see [Prefixed identifiers](#).

Some features added in Fix Pack 4:

- If you have existing Java code, you can now invoke it directly from ESQL. For more information, see [Java routines](#).
- The MRM Tagged/Delimited String format can now parse and write raw TLOG messages. For more information, see [TLOG messaging standard](#).
- The MRM Encoding Null mechanism has been extended for the COBOL initialization of 01 structures. For more information, see [Importing from COBOL: supported features](#).

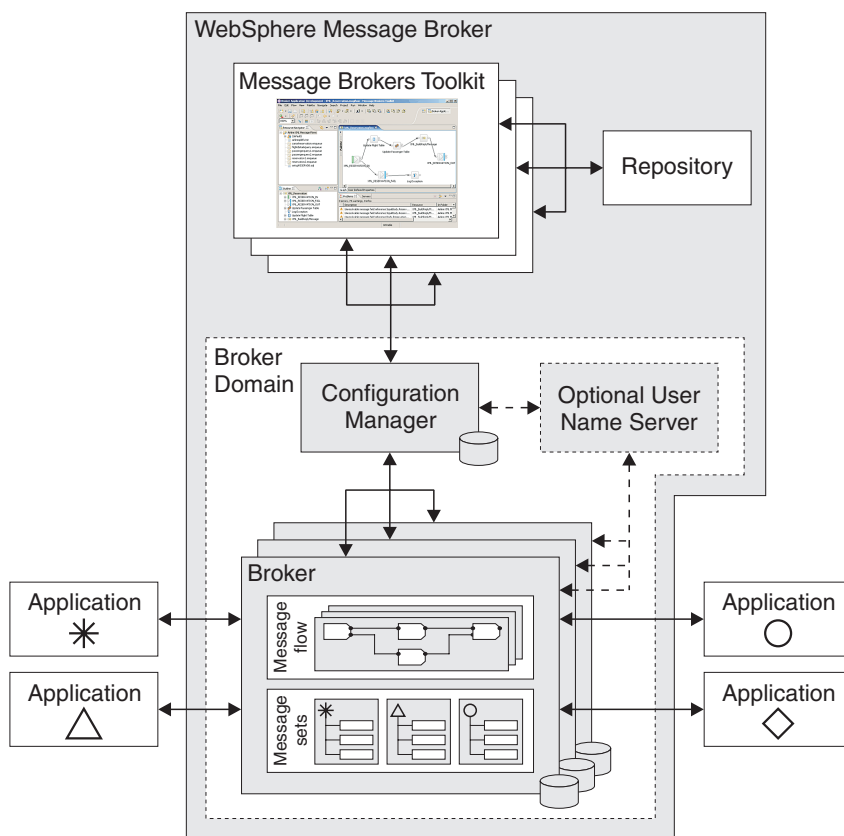
Some features added in Fix Pack 3:



- When an XML Schema is imported, the following constructs are now accepted: Key, Keyref, and Unique. However, XML Schemas with such constructs are not supported and are ignored by the broker. For more information, see Importing from XML Schema: unsupported features.
- The COBOL importer now supports the ASCII Sign EBCDIC Custom External Decimal representation within the CWF component. For more information, see Importing from COBOL: supported features.

## WebSphere Message Broker Version 6.0

WebSphere Message Broker enables information packaged as messages to flow between different business applications, ranging from large legacy systems through to unmanned devices such as sensors on pipelines.



There are two ways in which WebSphere Message Broker can act on messages.

### Message routing

Messages can be routed from sender to recipient based on the content of the message.

The message flows that you design control message routing. A message flow describes the operations to be performed on the incoming message, and the sequence in which they are carried out.

Each message flow consists of:

- A series of steps used to process a message. The steps are defined in message flow nodes.
- Connections between the nodes, defining routes through the processing. Connections are made using message flow node connections.

IBM supplies built-in nodes and samples for many common functions. If you require additional functions, you can write your own user-defined nodes.

You create message flows in the Message Brokers Toolkit; an integrated development environment and broker domain administration console.

## Message transformation

Messages can be transformed before being delivered:

- They can be transformed from one format to another, perhaps to accommodate the different requirements of the sender and the recipient.
- They can be transformed by modifying, combining, adding or removing data fields, perhaps involving the use of information stored in a database. Information can be mapped between messages and databases. More complex manipulation of message data can be achieved by using Extended SQL (ESQL).

Transformations can be made by various nodes in a message flow. But before a message flow node can operate on an incoming message, it must understand the structure of that message.

- Some messages contain a definition of their own structure and format. These are known as self-defining messages. Self-defining messages can be handled without the need for additional information about structure and format.
- Other messages do not contain information about their structure and format. To process them, a message definition of their structure must be created and made available.

The message definitions that you design are created within a message set, which contains one or more message definitions. Message sets also categorize message definitions. The category facility, which you can extend using XSLT scripts, is used for generating Web Services Description Language (WSDL) and documentation.

Like message flows, you create message definitions in the workbench. They can contain two types of information:

- The logical structure - the abstract arrangement and characteristics of the data, represented as a tree structure
- One or more physical formats - the way the data is represented and delimited in the physical bit stream

## Create the run time

The work of routing and transforming messages takes place in a broker. Brokers contain a number of execution groups; processes in which message flows are run.

Brokers are grouped into broker domains. Each domain is coordinated by a Configuration Manager. There can be many brokers, and each can be running on a different system. This provides protection against failure, and can separate work across different divisions in a business.

The system administrator creates the Configuration Manager with a command line instruction. The Configuration Manager uses an internal repository to store information relating to its broker domain.

The system administrator similarly creates one or more brokers, linking each to a particular Configuration Manager, thus making them part of the domain controlled by that Configuration Manager. Each broker uses a database to store the information it needs to process messages at run time.

The Configuration Manager also displays the users and groups in the Access control lists that you use to set user permissions, see *Publish/subscribe* below.

## Develop applications

After the system administrator has created and connected the components of the broker domain, an application developer creates and modifies message flows and message definitions using the workbench.

Different perspectives in the workbench are used to develop message flows and message sets as well as to administer one or more broker domains.

A repository can be used to provide access control and version control. A repository also allows multiple developers to work on the same resources in parallel.

You can use WebSphere MQ for communication between application and brokers. Other communication protocols you can use are:

- WebSphere MQ Enterprise Transport
- WebSphere MQ Mobile Transport
- WebSphere MQ Multicast Transport
- WebSphere MQ Real-time Transport
- WebSphere MQ Telemetry Transport
- WebSphere MQ Web Services Transport
- WebSphere Broker JMS Transport

## Deploy applications to the run time

When message flows and message sets have been created using the workbench, executable data can be deployed (transferred) to one or more brokers.

You can deploy data in one of two ways:

- From the workbench
- Using a shell command

When you deploy message flows and message sets, they are compiled and enveloped in a broker archive (bar) file, and sent to the Configuration Manager. The bar file has configurable system properties. You can override properties such as queue and database names, without the need to change source files or redevelop the message flow. This makes it easier to move definitions between systems.

The Configuration Manager opens the envelope, removes the contents, makes a record of the information that it has received, and routes the information to the appropriate brokers. (The envelope is discarded when the information it contains has been retrieved.) Each broker stores the information in its own local database.

This means that, when a broker has sufficient information, it can continue processing messages even if it is no longer connected to its Configuration Manager.

The Configuration Manager coordinates all activity (for example, changes to a message set) between the workbench and brokers within its domain. WebSphere MQ messaging is used between the workbench, the Configuration Manager, and the brokers.

## **Publish/subscribe**

The simplest way of directing messages is to use point-to-point messaging, by sending messages directly from one application to another. Publish/subscribe provides an alternative style of messaging.

A publishing application sends a message about a named topic to a broker. The broker passes the published message to those applications that have registered an interest in that topic. The publisher and the subscriber are unaware of the other's existence.

The broker handles the distribution of messages between publishing applications and subscribing applications. Applications can publish on, or subscribe to, many topics as well as apply more sophisticated filtering mechanisms.

An optional User Name Server in the broker domain controls who is authorized to publish or subscribe to topics. You set up and administer topic-based security from the workbench.

You set user permissions at individual or group level using Access control lists.

## **Message Brokers Toolkit**

The Message Brokers Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.

Application developers work in separate instances of the Message Brokers Toolkit to develop message sets and message flows. The Message Brokers Toolkit also communicates with one or more Configuration Managers, and you can use it to manage broker domains.

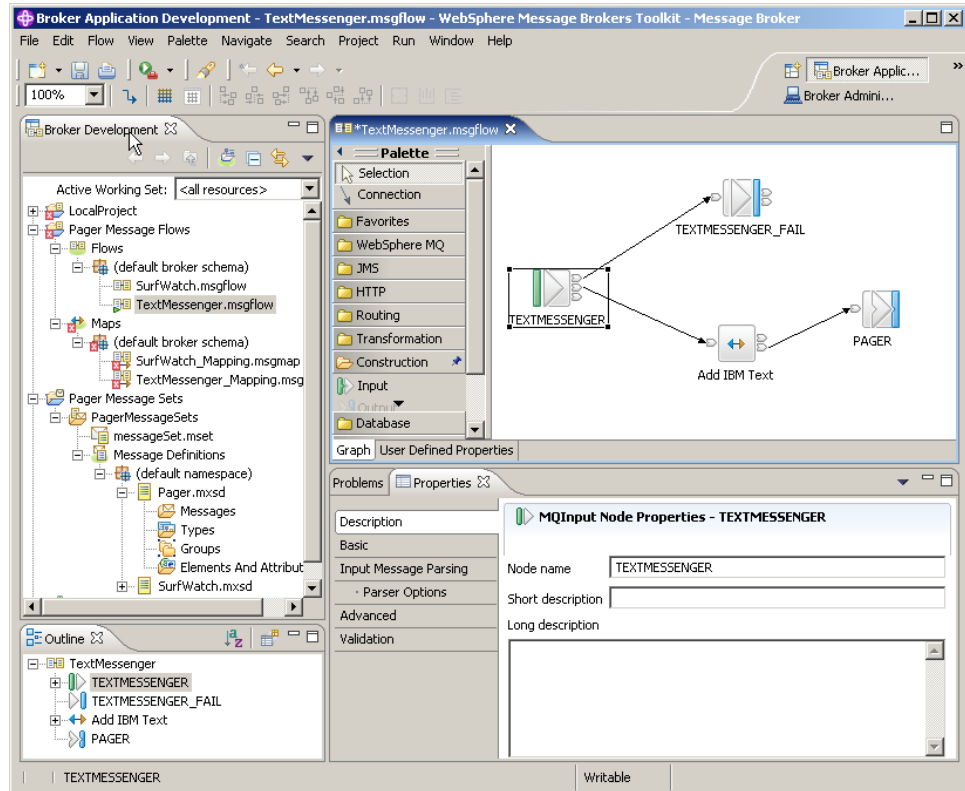
You can install the Message Brokers Toolkit on Windows and Linux (x86 platform).

### **The workbench**

When you start the Message Brokers Toolkit, a single window is displayed. This is the workbench, which contains one or more perspectives.

A perspective is a collection of views and editors that help you complete a specific task, or work with specific types of resource. The first time that you start the Message Brokers Toolkit, the Broker Application Development perspective is displayed. While you work in the workbench, you will switch perspectives frequently.

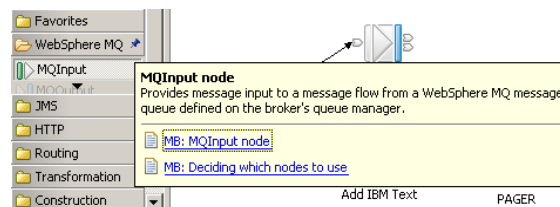
The following figure shows the Broker Application Development perspective with a message flow open in the Message Flow editor.



## Accessing context-sensitive help (infopops)

Infopops are provided throughout the Message Brokers Toolkit. An infopop is a small pop-up window that displays context-sensitive help and provides links to more information in the information center. You can launch an infopop for most aspects of the user interface (for example, on the Resource Navigator, the Message Flow editor, or a properties page) by bringing focus to the object and pressing F1 (on Windows) or SHIFT+F1 (on Linux).

The following figure shows the infopop that is displayed when you press F1 on the MQInput node in the node palette of the Message Flow editor.

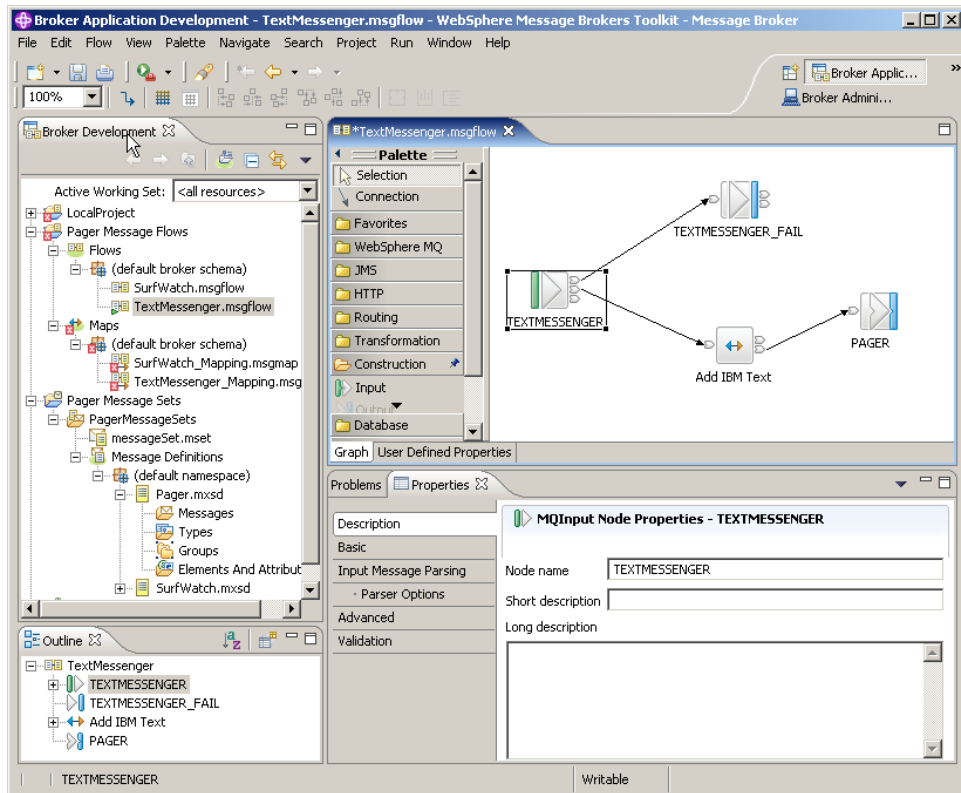


## Perspectives

A perspective is a group of views and editors that shows various aspects of the resources in the workbench, and helps you to complete specific tasks.

You can switch perspectives, depending on the task at hand, and customize the layout of views and editors. Switch between perspectives by clicking **Window** → **Open Perspective** → **Other** and clicking the name of the perspective to which you want to switch.

The following figure shows the Broker Application Development perspective in the Message Brokers Toolkit.



The Message Brokers Toolkit contains the following perspectives:

#### **Broker Application Development perspective**

The Broker Application Development perspective is the default perspective that is displayed the first time that you start the Message Brokers Toolkit. Application developers work in this perspective to develop and modify message sets and message flows. The figure above shows the Broker Application Development perspective with a message flow open in the Message Flow editor.

#### **Broker Administration perspective**

The Broker Administration perspective is a broker administration console that communicates with one or more Configuration Managers. Administrators work in this perspective to manage the resources (also referred to as domain objects) in the broker domain that are defined on one or more Configuration Managers.

#### **Debug perspective**

The Debug perspective is where application developers test and debug message flows.

#### **Plug-in Development perspective**

The Plug-in Development perspective is where application developers develop plug-ins for user-defined extensions.

#### **Data perspective**

The Data perspective is where application developers import relational database schemas for ESQL content assist and validation.

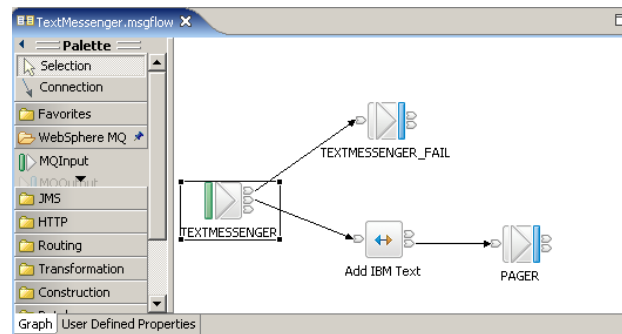
#### **Editors**

An editor is a component of the workbench. It is typically used to edit or browse a resource.

When you open a file for editing, for example by double-clicking it in the Broker Development view, the default editor associated with that file opens in the editor area of the current perspective. By default, the editor area is in the top-right hand side of the workbench window.

You are advised to open resources with the default editor only, because other editors might not correctly validate the changes you make.

The following figure shows the TextMessenger.msgflow file from the Pager samples opened in the Message Flow editor, which is part of the Broker Application Development perspective.



You can open any number of editors at once, but only one editor is active at any one time. The main menu bar and main toolbar display the operations that apply to the active editor. By default, editors are stacked in the editor area, but you can choose to tile them to view source files simultaneously. Tabs in the editor area indicate the names of the resources that are open for editing. An asterisk (\*) indicates that an editor has unsaved changes. If you attempt to close the editor or exit the workbench with unsaved changes you are prompted to save the changes.

## Resources

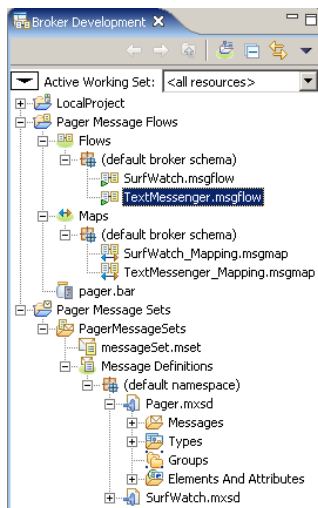
The projects, folders and files that exist in the workbench are called *resources*. They are collectively referred to as the *workspace* and they reside in your local file system.

By default, resources reside (with their metadata) in the workspace directory. The workspace directory is created the first time that you start the Message Brokers Toolkit. On Windows, the default workspace directory is created at C:\Documents and Settings\user\IBM\wmbt6.0\workspace, and on Linux, the default workspace directory is created at /home/user/IBM/wmqi6.0/workspace, where *user* is the user name with which you are logged on. You can create projects in other directories as well as in the workspace directory. You can maintain multiple workspaces by specifying a new location when prompted while the Message Brokers Toolkit is starting.

Typically, you open and view for editing any workbench resource in the Broker Development view. The exception is a broker domain resource (also referred to as a *domain object*) which you view and open for editing in the Broker Administration Navigator view in the Broker Administration perspective.

The following figure shows a server project, a message flow project, and a message set project in the Broker Development view of the Broker Application Development perspective.





**Types of resource:** There are three basic types of resource:

**Files** These are comparable to files in a file system. Different types of resource are maintained in different file types, for example, message flow files, message set definition files, ESQL files, mapping files, XML Schema files and broker archive files. File types are listed in “Resource types in the Message Brokers Toolkit” on page 112.

### Folders

These are comparable to directories in a file system. Folders are contained within projects or other folders. Folders can contain files and other folders.

When you create resources, they are grouped together by file type in logical categories within the hierarchy, and placed in virtual folders. Within a Message Flow project, there can be up to six virtual folders containing groups of resources. These are:

**Flows** This virtual folder is always shown after project creation, and contains .msgflow files, which are grouped by namespaces.

**Maps** This virtual folder is shown only when it contains .msmap files, which are grouped by namespaces.

### ESQLs

This virtual folder is shown only when it contains .esql files, which are grouped by namespaces.

### Database Connections

This virtual folder is shown only when it contains database related files, for example, .conxmi, .dbxmi, and .schxmi. These database files are grouped by namespaces.

### Enqueues

This virtual folder is shown only when it contains .enqueue files, which are grouped by namespaces.

### Flow Tests

This virtual folder is shown only when it contains .mbtest files, which are grouped by namespaces.

Within a Message Set project, there is a message set folder containing a message set file (.messageSet.mset) and potentially two virtual folders containing groups of resources. These are:



### Message Definitions

This virtual folder is always shown after project creation, and contains .mxsd files, which are grouped by namespaces.

### Message Categories

This virtual folder is shown only when it contains .category files, grouped by namespaces.

## Projects

Projects contain folders and files. Projects are used for building, version management, sharing, and resource organization. Like folders, projects map to directories in a file system. When you create a project, you specify a location for it in the file system. By default, projects are created in the workspace directory..

A project is either *open* or *closed*. You can view and modify an open project in the workbench. You cannot view or modify the contents of a closed project. The files and folders of a closed project are not displayed in the workbench, but they still reside on the local file system.

You can filter closed projects so that they do not appear in the workbench, by selecting **Filters** from the view menu (shown by a down arrow) in the Broker Development view, and then clicking **Closed Projects** from the displayed list of resource filters.

A closed project requires less memory than an open project.

You can:

- Add a project to the workbench.
- Save and version a project in an external repository.
- Copy a project from another user's workbench.

Project types are listed in "Resource types in the Message Brokers Toolkit" on page 112.

**Working sets:** A *working set* is a logical collection of application projects, that you can use to limit the number of resources that are displayed in the Broker Development view. Creating and using a working set allows you to reduce the visual complexity of what is displayed in the Broker Development view, making it easier to manage and work with your application projects.

The *active working set* is the current working set of application projects that you choose to display. Only the application projects contained in the active working set are displayed in the Broker Development view. If you do not create any working sets, the default active working set contains **<all resources>** and displays all of your application projects.

You can create a new working set in the Broker Development view in either of the following two ways:

- You can create a new working set and add existing application projects to it
- When you use a Quick Start wizard to set up the basic resources that are required to develop a Message Broker application, you are given the option of creating a new working set for the resources.

When you use the Broker Application Development perspective to create a new file or new project, the new file or project is automatically added to the current active

working set. If there is no active working set, the new file or project is displayed under **all resources** in the Broker Development view but is not added to a working set.

**By name linking:** Previously, objects such as nodes in a message flow, or an execution group, were identified by a system-generated identifier called a *universally unique identifier (uuid)*. When an object was removed, renamed or deleted, the reference to the object was lost.

In WebSphere Message Broker, users identify objects using a combination of a *namespace* and a name, referred to as a *fully qualified name*. The use of fully qualified names makes it easy to identify and locate objects, and to correct broken references. For example, if you rename an object, all references to that object are broken. If you substitute another object with the same name, all the broken references are corrected. This concept, called *by name linking*, is particularly important when you are working in a team environment. It means that you can share files in a repository and concurrently modify, add, and delete objects in your message flow application. When you integrate the various parts of the application, you can detect and resolve broken references to objects that have been moved, renamed or deleted.

**Project references:** A *project reference* is a property that you can set for a project so that it can be referenced by another project. When one project references another, the files in the referenced project are available for reuse.

For example, you might want to create a library of reusable ESQL subroutines in a project. Or you might want to create a similar library of message flows to reuse in other message flows. By adding a project reference to the library project, its subroutines or subflows are available for you to use.

Another use of project references is to enable Content Assist in the ESQL or Mapping editor. (“Content Assist” is context-sensitive help that displays valid ways in which a code statement can be completed.) For example, if you set up a project reference from a project containing ESQL code to a project containing a message set, the ESQL editor is able to display a list of valid message references.

If you subsequently close or delete a referenced project, or delete an object within it, it is no longer available to the referencing project and an error is generated. You can correct the error by opening the closed project or adding the missing object, with the correct name, and saving.

## Development repository

You can use any repository supported by Eclipse with WebSphere Message Broker. Many repositories provide features such as version control and access control of files, and make it easier for teams to work on shared resources.

You can take the version numbers of these artefacts out of the repository, and associate that version number with the message flows and message sets when they are deployed. This allows you to display which version of the flow is deployed in the workbench. An alternative method of assigning a version number to a message flow is to specify one in the workbench when the message flow is created, see Message flow version and keywords.

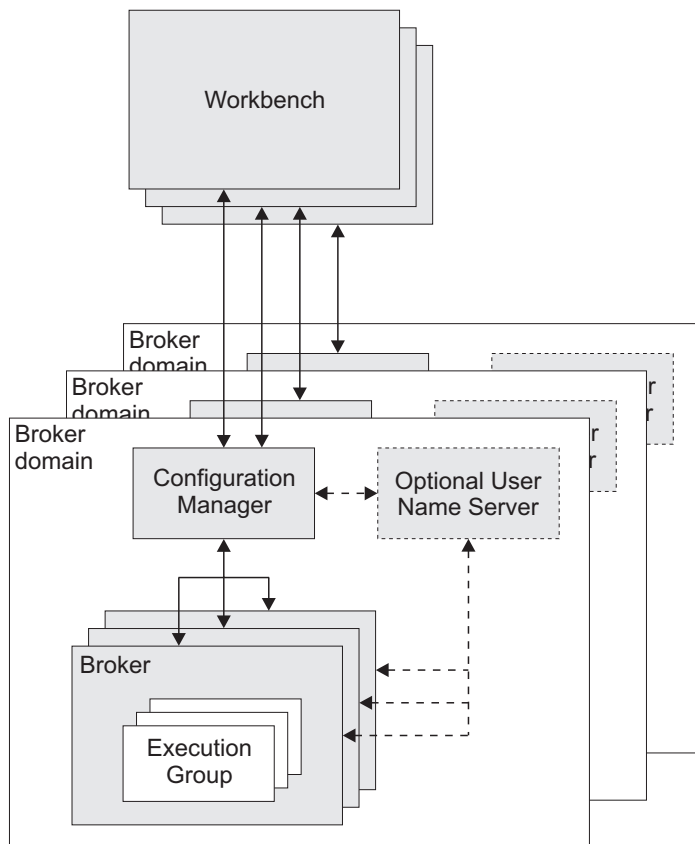
Because the Message Brokers Toolkit is based on the Eclipse platform, you can access Eclipse-supported repositories directly from the Message Brokers Toolkit.

Configuring CVS to run with the Message Brokers Toolkit describes how you set up the Message Brokers Toolkit to run with CVS. For more information about other repositories, refer to the section "Repository Providers" in the Eclipse developer community.

## Runtime environment

The runtime environment is the set of resources that exist at run time, including the Configuration Manager, brokers, and execution groups.

The following figure shows the relationship between the resources that exist at run time, and how they interact with the workbench.



Follow the links below for more information about each of the runtime components:

- "Brokers" on page 70
- "Execution groups" on page 72
- "Broker domains"
- "Configuration Manager" on page 68
- "User Name Server" on page 74

### Broker domains

A broker domain is one or more brokers that share a common configuration, together with the single Configuration Manager that controls them.

You install, create, and start one or more brokers, and an optional User Name Server, in a broker domain. You can configure more than one broker domain, each managed by its own Configuration Manager.

The Configuration Manager uses an internal repository to store and share information about the resources and components in the domain.

You create the physical components of a broker domain using command line instructions, and then configure and administer the broker domain using the Broker Administration perspective in the workbench.

## Configuration Manager

The Configuration Manager is the interface between the workbench and an executing set of brokers. It provides brokers with their initial configuration, and updates them with any subsequent changes. It maintains the broker domain configuration. The Configuration Manager is supported on all the broker platforms.

The Configuration Manager is the central runtime component that manages the components and resources that constitute the broker domain.

The Configuration Manager has four main functions:

- Maintains configuration details in an internal repository. This internal repository provides a central record of the broker domain components.
- Deploys the broker topology and message processing operations in response to actions initiated through the workbench. Broker archive (bar) files are deployed through the Configuration Manager to the execution groups within a broker.
- Reports on the results of deployment and the status of the broker.
- Communicates with other components in the broker domain using WebSphere MQ transport services.

You must install, create, and start a Configuration Manager for each broker domain. You can install, create, and start more than one Configuration Manager on one or more machines.

When you create a Configuration Manager, you must give it a name that is unique within the broker domain; names cannot be shared between Configuration Managers, or between Configuration Managers and brokers.

You can install and configure the Configuration Manager runtime component on AIX, HP-UX, Linux, Solaris, Windows, and z/OS. For more information about supported platforms see Operating system requirements .

You administer one or more Configuration Managers in the Broker Administration perspective in the workbench. More than one instance of the workbench can access a Configuration Manager concurrently.

The Configuration Manager can share a host queue manager with one broker in the broker domain. To communicate with other brokers in the broker domain, the Configuration Manager requires sender and receiver channels.

**Resources associated with a Configuration Manager:** When you create the Configuration Manager runtime component, the following resources are also created:

- An internal repository.

- A WebSphere MQ queue manager (unless one exists already) on the same physical system as the Configuration Manager. A queue manager can be shared with a single broker.
- A set of fixed-name queues, defined to the queue manager that hosts the Configuration Manager.

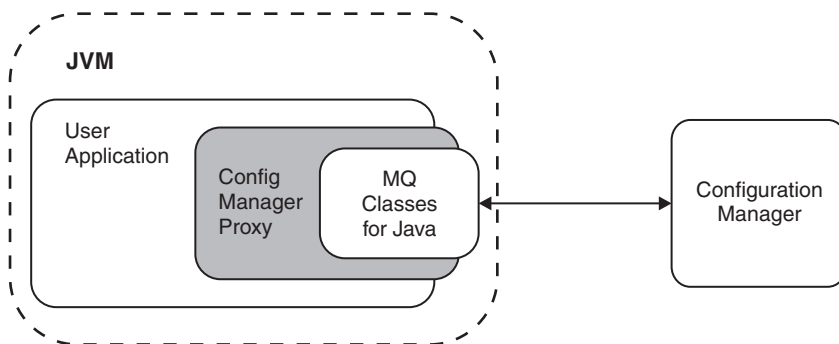
### Configuration Manager Proxy:

The Configuration Manager Proxy (CMP) is an application programming interface that your applications can use to control broker domains through a remote interface to the Configuration Manager.

Your applications have complete access to the Configuration Manager functions and resources through the set of Java classes that constitute the CMP. For example, you can use the CMP to interact with the Configuration Manager to:

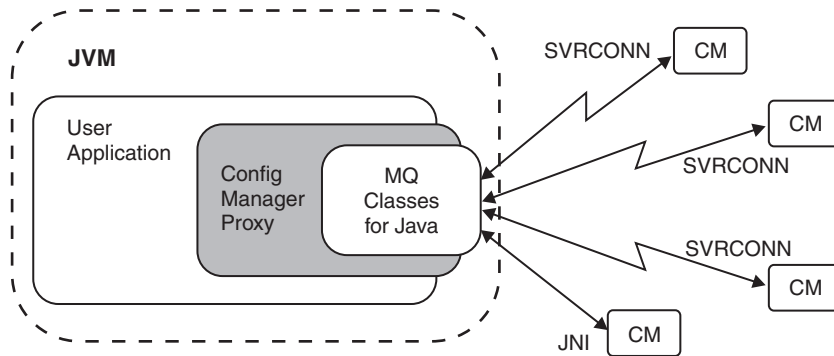
- Deploy BAR files, Publish/Subscribe topology, topic trees, and broker configuration.
- Modify the Publish/Subscribe topology; add and remove brokers, broker connections, and collectives.
- Create, modify, and delete execution groups
- Enquire and set status of objects in the domain (for example, run state), and be informed if status changes.
- Manipulate the topics hierarchy.
- View the broker event log and active subscriptions table.
- Modify domain Access Control Lists, when connected to a Version 6.0 Configuration Manager only.

The CMP is a set of Java classes that sit logically between the user application and the Configuration Manager, inside the Java Virtual Machine (JVM) of the user application. It requires the WebSphere MQ Classes for Java in order to function, as shown below.



The CMP application can be on the same physical machine as the Configuration Manager, connected by a JNI (Java Native Interface) connection to the queue manager using the WebSphere MQ Java Bindings transport. Alternatively, it can be distributed over a TCP/IP network, and connected to the queue manager by a WebSphere MQ SVRCONN channel using the WebSphere MQ Java Client transport.

You cannot use the CMP to communicate with more than one Configuration Manager from within the same application:



Using the API, you can connect to and manipulate a Configuration Manager in the following products:

- WebSphere Business Integration Event Broker Version 5.0
- WebSphere Business Integration Message Broker Version 5.0
- WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0
- WebSphere Event Broker Version 6.0
- WebSphere Message Broker Version 6.0

A domain controlled by a Version 5.0 Configuration Manager can consist of Version 2.1 and Version 5.0 brokers, to which either version can be deployed by the CMP. Although you can run only one Version 5.0 Configuration Manager on each physical machine, a single CMP application can still connect to multiple Version 5.0 Configuration Manager components.

## Brokers

A broker is a set of execution processes that hosts one or more message flows to route, transform, and enrich inflight messages.

Applications send messages to the broker using WebSphere MQ queues and connections. The broker routes each message using the rules defined in message flows and message sets, and transforms the data into the structure required by the receiving application.

The broker uses sender and receiver channels to communicate with the Configuration Manager and other brokers in the broker domain.

You install the broker component on the platforms described in Operating system requirements. If you have installed the broker component, you can also install the Message Transformation Services, which are a set of message processing services that extend the capabilities of the broker. For example, the Message Transformation Services provide additional message processing nodes, additional input and output nodes, and message flow debug capabilities. You create the broker using command line instructions on the machine where the component is installed. You can install and create one or more brokers, on one or more machines.

The broker depends on a broker database to hold broker information. This includes control data for resources defined to the broker, for example deployed message

flows. You need to define a database and authorize access for specific users before you create the broker because creating the broker creates tables within the database. The database is also known as the broker's local persistent store.

The broker connects to the database using an ODBC connection.

When you create a broker, you must give it a name that is unique within the broker domain. Broker names are case-sensitive on all supported platforms, except Windows platforms. You must use the same name when you create a reference to the broker in the broker domain topology in the workbench. The reference to the broker is a representation of the physical broker in the configuration repository.

When you have created the broker reference, you deploy the changes to your broker domain. Deployment starts communications between the broker and the Configuration Manager. The broker receives configuration information from the Configuration Manager, and stores it in the configuration repository. Deployment also initializes the broker to make it ready to execute message flows.

**Resources associated with a broker:** When you create a broker, the following resources are also defined and created:

- A set of database tables for storing the information used by the broker to process messages at run time.

You can use one of a number of database products to create the database tables, depending on which platform the product is installed. See Supported databases for more information.

**z/OS** You must create a DB2 database if you are using WebSphere Message Broker for z/OS.

- A WebSphere MQ queue manager, if one does not already exist.
- A set of fixed-name queues that are defined to the WebSphere MQ queue manager.

### **System management interfaces:**

The brokers provide a service for independent system management agents.

This enables a central management facility to access information about any network that includes a WebSphere Message Broker broker domain.

This support ensures that existing system management agents can be extended to include WebSphere Message Broker resources.

WebSphere Message Broker brokers publish event messages, using fixed topics, in response to configuration changes, state changes, and user actions such as subscription registrations. WebSphere Message Broker also uses architected messages to publish events related to the status, and change in status, of the brokers. These messages are published using the reserved topic root `$$SYS` in code page 1208.

An example of a fixed topic is:

```
$$SYS/Broker/<brokerName>/Status/ExecutionGroup/<executionGroupName>
```

So the topic structure is fixed in this case, but obviously `<brokerName>` and `<executionGroupName>` are replaced with the appropriate values.



An example of the actual message data for this publication is:

```
<Broker uuid="12345678-1234-1234-1234-123456789012">  
  <ExecutionGroup uuid="12345678-1234-1234-1234-123456789012">  
    <Stop>  
      <AllMessageFlows/>  
    </Stop>  
  </ExecutionGroup>  
</Broker>
```

A system management agent can subscribe to these topics, or to a subset of these topics, to receive the detailed information about activity and state changes in the WebSphere Message Broker broker domain.

The event messages have a fixed structure, defined in *XML (Extensible Markup Language)*. The format of these messages, constructed in XML, is detailed in The XML message body. The messages cover configuration changes, state changes, error notifications, and detailed subscription and topic information (for example, a subscription registration).

You can develop or purchase system management adapters or customized administrative applications. These subscribe to the system management topics generated by WebSphere Message Broker to receive information on the broker domain activity.

### **Execution groups:**

An execution group is a named grouping of message flows that have been assigned to a broker. The broker enforces a degree of isolation between message flows in distinct execution groups by ensuring that they execute in separate address spaces, or as unique processes.

Each execution group is started as a separate operating system process, providing an isolated runtime environment for a set of deployed message flows. Within an execution group, the assigned message flows run in different thread pools. You can specify the size of the thread pool (that is, the number of threads) that are assigned for each message flow by specifying the number of additional instances of each message flow.

A single default execution group is set up ready for use when you create a reference to a broker in the workbench. By setting up additional execution groups, you can isolate message flows that handle sensitive data such as payroll records, or security information, or unannounced product information, from other non-sensitive message flows.

If you create additional execution groups, you must give each group a name that is unique within the broker, and assign and deploy one or more message flows to each one.

You can create and deploy execution groups either in the workbench, or using commands.

An execution group process is also known as a DataFlowEngine (DFE); this term is typically used in problem determination scenarios (trace contents, diagnostic messages, and so on). A DFE is created as an operating system process, and has a one-to-one relationship with the named execution group. If more than one message flow runs within an execution group, multiple threads are created within the DFE process.



## 64-bit execution groups

When you create an execution group, you can specify whether you want to create a 32-bit execution group or a 64-bit execution group.

### Why create 64-bit execution groups?

There are two main benefits to creating 64-bit execution groups:

- If you are using both a 64-bit queue manager and a 64-bit execution group, you can run the broker as a WebSphere MQ trusted application, which means that the broker can handle WebSphere MQ operations faster than a normal broker. You cannot run a broker as a trusted application if you have a 64-bit broker queue manager but only a 32-bit execution group.
- You can allocate large amounts of contiguous memory. Message flows that are deployed to a 64-bit execution group can process larger messages than message flows that are deployed to a 32-bit execution group.

### Supported platforms

You can create 64-bit execution groups only on the following UNIX systems:

- AIX
- Solaris (SPARC platform)
- HP-UX (PA-RISC platform)
- HP-UX (Integrity platform)

When you deploy the first message flow to a new 64-bit execution group, the broker determines whether a 64-bit execution group is supported on that broker; if not, the deployment fails with error message BIP2825.

See Enabling connections to the databases to check whether you need to define 32-bit data source names (DSNs), 64-bit DSNs, or both when you are using a 64-bit execution group or when you configure global coordination of message flows that are deployed to a 64-bit execution group.

### Considerations when using 64-bit execution groups

The following list describes the main differences between using 32-bit and 64-bit execution groups:

#### *Execution group*

You must create a new execution group and specify, at creation time, that it will be a 64-bit execution group. You cannot change an existing 32-bit execution group into a 64-bit execution group.

#### *Broker queue manager*

The broker's queue manager must be 64-bit.

#### *Databases*

If you use a DB2 database, it must reside in a 64-bit DB2 instance. With Oracle, Sybase, Informix, and SQLServer, you can use 32-bit or 64-bit databases depending on how you connect the broker system to the database server.

### *User-defined nodes*

You must recompile as 64-bit applications any 32-bit user-defined nodes. For more information, see *Installing a user-defined extension to current and past versions of the broker*.

### *Message flows*

When you deploy a message flow to an execution group, the message flow runs as either a 32-bit application or a 64-bit application depending on whether the execution group is 32-bit or 64-bit. You do not need to change existing message flows to run in a 64-bit execution group unless the message flow includes a user-defined node that is compiled for a 32-bit application.

### *Flow debugger*

The flow debugger is not supported for use with message flows that are deployed to 64-bit execution groups.

## **User Name Server**

The User Name Server is an optional runtime component that provides authentication of users and groups performing publish/subscribe operations.

If you have applications that use the publish/subscribe services of a broker, you can apply an additional level of security to the topics on which messages are published and subscribed. This additional security, known as topic-based security, is managed by the User Name Server. It provides administrative control over who can publish and who can subscribe. For example, if a client application publishes messages containing sensitive company finance information, or personnel details, the User Name Server can be used to restrict access to those messages.

The User Name Server interfaces with operating system facilities to provide information about valid users and groups in a broker domain.

You install, create, and start a User Name Server in any supported operating environment. These are listed in *Supported processors and Operating system requirements*.

The User Name Server can share a host queue manager with the Configuration Manager and one broker in the broker domain. To communicate with other brokers in the broker domain, the User Name Server requires sender and receiver channels.

**Resources associated with a User Name Server:** When you create a User Name Server, the following resources are also created:

- A WebSphere MQ queue manager unless one exists already. The queue manager is created on the same physical system as the User Name Server. A queue manager can be shared with the Configuration Manager, a single broker, or both.
- A set of fixed-name queues, defined to the WebSphere MQ queue manager that hosts the User Name Server.

---

## **WebSphere Message Broker scenario**

WebSphere Message Broker manages the flow of information in your business, applying business rules to route, store, retrieve, and transform the information as required by the different systems in your business and the systems of your customers, suppliers, partners, and service providers.

The following scenario outlines an example of how WebSphere Message Broker can be used to solve IT infrastructure problems in a business.

## Mergers and acquisitions scenario

This scenario describes how WebSphere Message Broker is used by a fictitious insurance company to manage two disparate IT infrastructures after a small, Internet-based insurance company is acquired by a large, more traditional insurance company. The description focuses on what happens when a potential customer requests a motor insurance quotation using the merged company's Web site. This scenario is based on a larger, more complex scenario that was published on developerWorks. To read the full scenario see the links at the end of the scenario description.

### Background

Company A is a motor and general insurance company that has been in business for about 50 years and currently has about 5 million policyholders. The company uses agents and a call center to communicate with customers. The company has a large legacy IT infrastructure, which includes CICS Transaction Server on z/OS and IBM DB2 Universal Database on z/OS.

Company B is small Internet-based motor insurance company, which currently has less than one million policyholders but is expanding. The company's IT infrastructure includes WebSphere Application Server on Windows 2000, and Oracle Enterprise and IBM DB2 Universal Database on Windows 2000.

### Problems

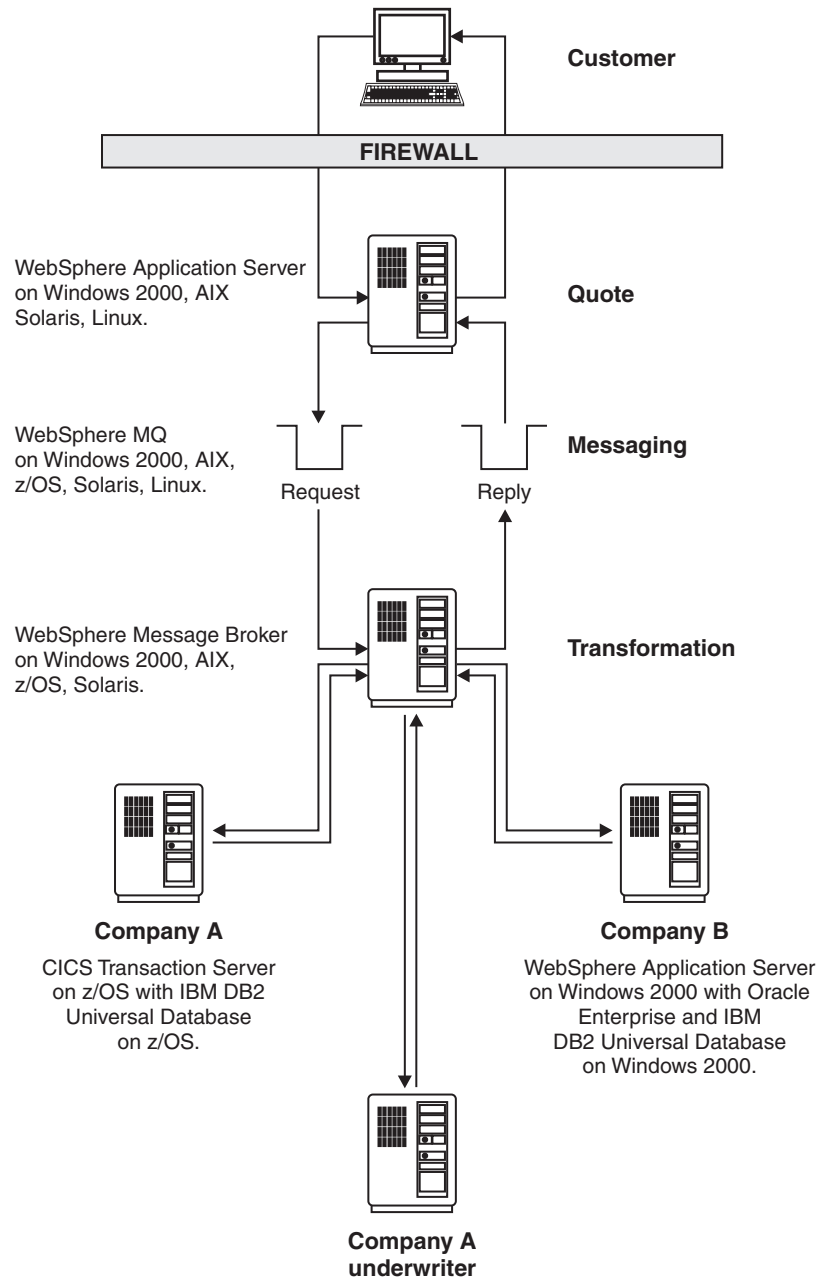
Company A acquired Company B to gain access to the Internet-based insurance market and to use Company B's Internet-based skills and IT infrastructure. The two companies have customer and policy data of different formats but, for legal reasons, the data from the separate companies cannot be merged. However, the administration costs of managing the separate IT infrastructures are high. Also, customers, agents, and call center staff need to have a single administration process to interact with the company's data.

### Solution

Now that the two companies have merged, users can request an insurance quotation by giving some basic personal information in a form on the new company's Web site. WebSphere Application Server, on which the Web site runs, forwards the request in XML format to WebSphere Message Broker using the request queue in a WebSphere MQ cluster. WebSphere Message Broker transforms the XML request to the legacy COMMAREA format that is used by Company A's systems, then routes the request to Company A's systems. WebSphere Message Broker also routes the request, in XML format, to Company B's systems. Both systems return a quotation to WebSphere Message Broker.

Logic within WebSphere Message Broker also requests a risk assessment from the internal underwriter and applies the returned risk to the quotations from Company A's and Company B's systems. The broker detects that, in this instance, the best or lowest quotation for the customer has been generated by Company A's systems. So the broker transforms Company A's quotation from COMMAREA to XML and routes the quotation back to WebSphere Application Server using a reply queue in the WebSphere MQ cluster where the quotation is stored for up to 14 days. WebSphere Application Server returns the quotation to the customer.

The following diagram illustrates the flow of information in this scenario.



## Accessibility features for WebSphere Message Broker

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

### Accessibility features

The following list includes the major accessibility features in WebSphere Message Broker. You can use screen-reader software to hear what is displayed on the screen.

- Supports keyboard-only operation
- Supports interfaces commonly used by screen readers

**Tip:** The WebSphere Message Broker Information Center, and its related publications, are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

## **Keyboard navigation**

This product uses standard Microsoft® Windows® navigation keys.

Visit the *IBM Accessibility Center* for more information about IBM's commitment to accessibility.



---

## Part 3. Reference

<b>Workbench</b> . . . . .	81
Perspectives in the Message Brokers Toolkit . . . . .	81
Broker Administration perspective. . . . .	81
Broker Application Development perspective . . . . .	83
Debug perspective . . . . .	87
Plug-in Development perspective . . . . .	90
Editors in the Message Brokers Toolkit . . . . .	90
Editor preferences and localized settings. . . . .	91
Broker Archive editor . . . . .	91
Broker Topology editor . . . . .	93
ESQL editor . . . . .	93
Event Log editor . . . . .	96
Message Category editor . . . . .	97
Message Definition editor . . . . .	98
Message Flow editor . . . . .	102
Message Mapping editor . . . . .	107
Message Node editor. . . . .	108
Message set editor. . . . .	109
Subscriptions Query editor . . . . .	110
Topics Hierarchy editor . . . . .	111
Resource types in the Message Brokers Toolkit . . . . .	112
Message flow projects and files . . . . .	112
Message set projects and files . . . . .	113
Plug-in Development projects and files . . . . .	114
Rules for naming workspace objects . . . . .	115
Message Brokers Toolkit keyboard shortcuts . . . . .	115
<b>Glossary of terms and abbreviations.</b> . . . . .	119





---

## Workbench

The Message Brokers Toolkit integrated development environment provides your application development and administration workbench on Windows and Linux (x86 platform). Within the workbench, you can access resources through several perspectives, and create and modify them using the supplied editors:

- Perspectives
- Editors
- Resource types

For information about how to use the keyboard in the workbench, see Keyboard shortcuts.

When you create resources in the workbench, or import resources into the workbench, be aware of the file system limit of 256 characters that is imposed by the Windows file system. This limit can cause restrictions in path specifications to resources (for example, message flow projects), and might cause access problems if the combination of path and resource name exceeds this limit. Keep installation locations and resource names short to avoid problems associated with this restriction.

A minimum display resolution of at least 1024 x 768 is required for some dialogs (for example, the Preferences dialog).

---

## Perspectives in the Message Brokers Toolkit

Follow the links below for a description of each of the perspectives in the Message Brokers Toolkit:

- “Broker Administration perspective”
- “Broker Application Development perspective” on page 83
- “Debug perspective” on page 87
- “Plug-in Development perspective” on page 90

### Broker Administration perspective

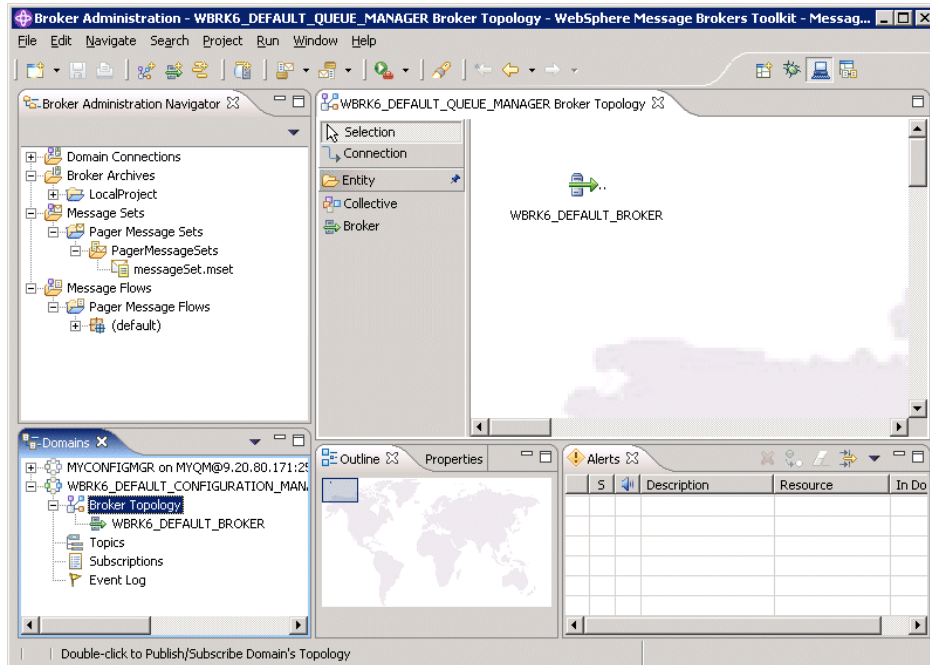
This is where the broker domain resources (also referred to as *domain objects*) that are defined on one or more Configuration Managers are managed.

Typically, you carry out the following tasks in this perspective:

- Setting up the broker domain, for example creating collectives
  - Creating and removing domain connections.
  - Connecting and removing brokers.
- Adding and removing execution groups.
- Creating and deploying broker archive (bar) files.
- Defining and managing publish/subscribe topic hierarchies.
- Querying, viewing and deleting subscriptions.
- Clearing and filtering deploy event log entries.
- Activating, deactivating, filtering, and clearing alerts.

- Adding, removing, and updating entries in access control lists (ACLs).

The following figure shows the Broker Administration perspective of the Message Brokers Toolkit. The Domains view in the figure shows that the Message Brokers Toolkit is connected to the Default Configuration.



The Broker Administration perspective provides several views that allow you to navigate and update your broker domain resources.

## Broker Administration Navigator view

This is where you view and work with any of the broker domain resource files. You can also deploy and delete any resource file that has been created in the Broker Application Development perspective.

The broker domain resource files are contained in the following folders:

### Domain Connections

Contains the domain connections that are defined to multiple local and remote Configuration Managers.

### Broker Archives

Contains the broker archive (bar) files.

### Message Sets

Contains the message set (mset) files.

### Message Flows

Contains the message flow (msgflow) files.

## Domains view

This is where you view and work with any of the broker domain objects listed below, when a broker domain is connected to a Configuration Manager:

- Broker Topology
- Topics

- Subscriptions
- Event Log entries

You can manage deployed configurations in this view. Double-clicking an object launches the appropriate editor in the editor view. You can also deploy bar files on execution groups, and start and stop message sets and message flows.

By default, an alert icon is displayed for a domain object that is connected to a Configuration Manager but is not in a normal running state. You can choose not to display the alert icon using an option that is available in the Alerts view.

## Editor view

You can open any broker domain resource to launch its default editor in the editor view. The following default editors are available:

### Broker Topology editor

For administering the brokers and collectives within a broker domain.

### Topics Hierarchy editor

For managing topics within a publish/subscribe system.

### Subscriptions Query editor

For viewing and deleting subscriptions within a publish/subscribe system. Subscriptions that use the WebSphere MQ Real-time Transport are not displayed in the Subscriptions Query editor.

### Event Log editor

For viewing, clearing and filtering event log entries.

## Properties view

Displays the property names and values for the domain object selected in the Broker Administration Navigator view.

## Alerts view

An alert is displayed in the Alerts view when a domain object is connected to a Configuration Manager but is not in a normal running state. The normal running state for a domain object is *started*, without a user trace or any other special processing enabled.

The Alerts view is updated continuously while the workbench is connected to a Configuration Manager.

You can filter, clear, and remove alerts from this view. You can also quieten alerts. This removes the alert icon from every object in the hierarchy in the Domains view, except for the alerted object and its parent.

When a new object is added to the domain it is selected in the Alerts Filter by default. Therefore, you might want to modify the selection of objects in the Alerts Filter after a new object has been added to the domain by you or anyone else.

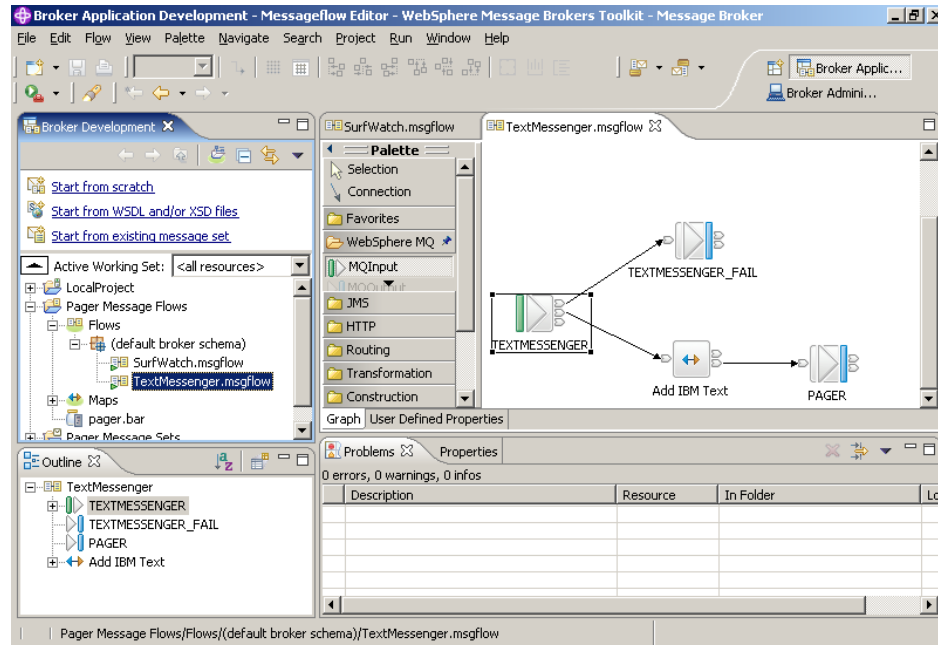
## Broker Application Development perspective

The Broker Application Development perspective is the default perspective that is displayed when you start the Message Brokers Toolkit.

Typically, you perform the following tasks in this perspective:

- Develop message sets and message flows.
- Enqueue and dequeue test and production messages, which is particularly useful when you are debugging message flows.


The following figure shows the Broker Application Development perspective of the Message Brokers Toolkit. The TextMessenger.msgflow file in the Broker Development view is open in the Message Flow editor.



The Broker Application Development perspective provides several views to navigate, browse, and update your application resources.

## Broker Development view

You use the Broker Development view to navigate and browse your application resources, and to select resources for editing. You can also use this view to access the Quick Start wizards, which help you create the basic resources needed for a Message Broker application.

At the top of the Broker Development view (on the left-hand side), you can click on a down arrow  to display a list containing links to the three **Quick Start** wizards:

- Start from scratch
- Start from WSDL and/or XSD files
- Start from existing message set

The Broker Development view displays a hierarchical view of the following message flow and message set resource files:

- Message flow projects
- Message flows
- ESQL files
- Mappings files
- Message set projects

- Message sets
- Message definition files

When you create resources, they are grouped together by file type in logical categories within the hierarchy, and placed in virtual folders. These virtual folders are described in “Resources” on page 63.

Other folders and files that are created for the Message Flow project or Message Set project are placed directly under the project folders.

You can hide the categories (virtual folders) by clicking the **Hide Categories** button in the Broker Development view toolbar, or by right-clicking the Message Flow project or Message Set project and then selecting **Hide Categories** from the menu. You can use the **Hide Categories** menu item or button to toggle between category mode and non-category mode.

You can also toggle between showing and hiding broker schema (in a Message Flow project) and namespaces (in a Message Set project) by right-clicking on the project and selecting **Hide Namespaces** from the menu. You can also do this by selecting **Hide Namespaces** from the view menu (shown by a down-arrow) in the Broker Development view toolbar. By default, broker schema and namespaces are shown. This option is available only in category mode.

You can toggle between showing and hiding file extensions for files that are under virtual folders, broker schema (or namespaces) and messageSet.mset. To toggle between hiding and showing the file extensions, either right-click on the file and then select **Hide Categorized File Extensions** from the menu, or select **Hide Categorized File Extensions** from view menu (shown by the down arrow) in the Broker Development view toolbar. By default, file extensions are shown. Other files and folders that are directly under the project are not affected by this option.

By default, the hierarchy view contains the resource files for all of your application projects. However, you can limit the number of resources that are displayed in the Broker Development view by organizing and displaying them in *working sets*. A working set is a logical collection of application projects, which makes it easier to manage and work with your application projects. For more information on working sets, see “Resources” on page 63 or Displaying selected projects in working sets.

You use specialized editors to open and edit any of the resources in the broker development view, except project files.

## Editor view

Double-click a resource in the **Broker Development** view to launch the associated editor in the Editor view.

The Message Brokers Toolkit provides the following specialized editors by default for working with the content of message flow and message set files:

### ESQL editor

For viewing and updating the ESQL statements that are associated with Compute, Database, and Filter nodes.

### Message Category editor

For creating, viewing, and updating message categories that you define within the MRM domain.

### Message Definition editor

For creating, viewing, and updating messages that you define within the MRM domain.

### Message Flow editor

For creating, viewing, and updating message flows, and the nodes that define them.

### Message Mapping editor

For defining transformations between data sources and targets, without the need for programming in XPath, XSLT, XQuery, Java, or ESQL.

### Message Set editor

For creating, viewing, and updating message sets that you define within the MRM domain.

## Outline view

The Outline view provides a summary of the content of the resource that is currently open in the Editor view.



Click the Overview Mode icon to toggle to the Overview view, which provides a useful summary for large, complex message flows because it shows a small-scale version of the flow. Click the Overview Mode icon again to return to the Outline view.

## Problems view

The Problems view displays any messages (information and error) that are associated with the resource that is currently open in the Editor view. For example, if you save a message flow that has an error (such as a mandatory property not set), you can check the content of the Problems view to determine any corrections that you need to make. When you double-click a Problems view entry, the appropriate editor opens the resource that contains an error, and positions the cursor at the point of the error (where possible).


## Properties view


The Properties view displays the properties for the node that is selected in the Editor view. The Properties are grouped and displayed on tabs, which are listed on the left of the Properties view. If you prefer to have the nodes' properties displayed in the Properties dialog box:

1. Click **Window** → **Preferences**.
2. Expand **Broker Development**, and click **Message Flow Editor**.
3. Select **Show Node Properties in a Properties Dialog**.

## Broker Application Development perspective toolbar












A new Broker Application Development perspective toolbar displays only the icons for **Get a message from a queue** and **Put a message to a queue**, shown in the table below.

Icon	Label and action
	Get a message from a queue

Icon	Label and action
	Put a message to a queue

You can add more icons to the toolbar for other actions that you might want to perform in the perspective. To add icons to the toolbar, do the following:

1. Click **Window** → **Customize Perspective...**. The Customize Perspective dialog is displayed.
2. Click the **Commands** tab. This tab displays the available command groups and their corresponding menu bar details and toolbar details.
3. Click a command group name to display its toolbar details. The available command groups and toolbar details for the icons that you might want to use in the perspective are shown in the table below.
4. In the broker, select the command groups that you want to add, and click **OK**. The toolbar displays the icons from the selected command groups.

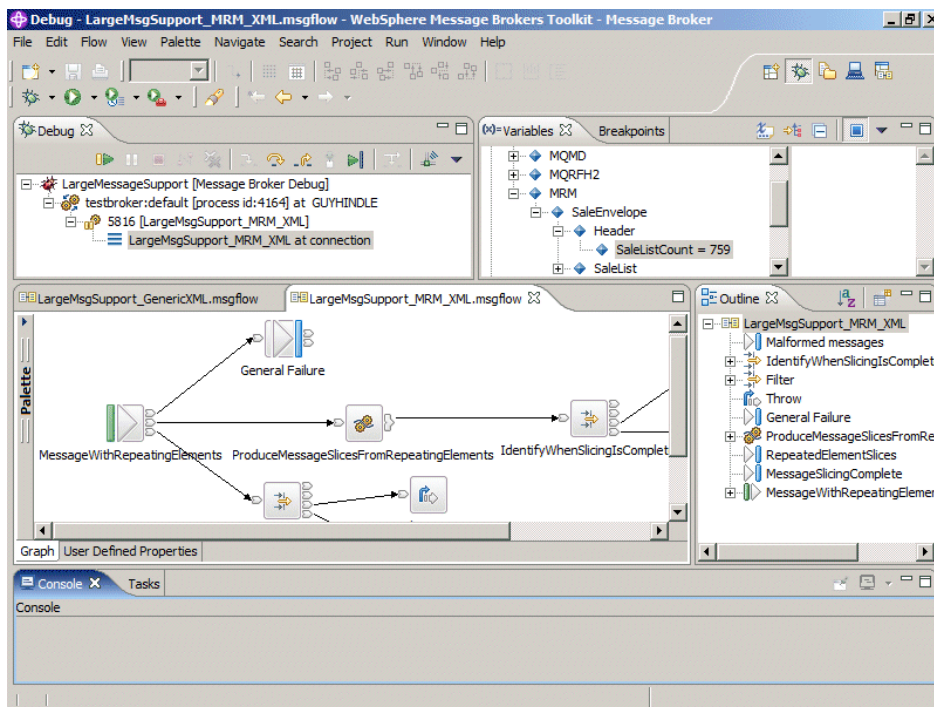
Available command groups	Toolbar details: icon	Toolbar details: label and action
ESQL Actions		Generate a New Message Flow ESQL File
Mapping Actions		Create a Message Map File
Message Creation		Create a New Message Set
		Create a New Message Definition File
		Create a New Message Category File
Message Flow Element Creation		Create a New Message Flow Project
		Create a New Broker Schema
		Create a New Message Flow
Message Set		Generate a Web Service Definition File
		Generate HTML Documentation for a Message Set
		Generate an XML Schema from a Message Definition File

## Debug perspective

The Debug perspective is where you test and debug a graphical representation of your message flows using the message flow debugger.

The following figure shows the Debug perspective of the Message Brokers Toolkit. In the figure, the LargeMsgSupport\_MRM\_XML message flow is being debugged.





## Debug perspective views

The Debug perspective contains the following views:

### Debug view

Displays the deployed message flow types for a selected host to help you manage flow debugging. Toolbar buttons are provided for controlling the execution of the flow. You can start, stop, and resume a flow, step into, and out of a subflow, and step into the source code.

When you attach the flow debugger to the flow runtime engine, the Debug view displays the names of the following flow-related entities:

- The host computer and the flow runtime engine that it is running. This is shown as a concatenation of the names of the following entities, delimited by colons.
  - The host computer
  - The broker
  - The execution group
  - The flow engine


The entry is identified by the flow runtime engine symbol. For example:

–  TestPC01:WMQIV5BR:TestExecution:DataFlowEngine

- The flows that are deployed in the flow runtime engine, identified by the flow symbol, for example:

–  TestFlow

- When a breakpoint is reached, the flow instances that have been created for each flow, identified by the following symbol, for example:

–  3068 (Paused)

In the Debug view, you can perform the following debugging tasks:

- Query a flow runtime engine for currently deployed flows
- Detach a flow runtime engine from the flow debugger





- Resume flow execution
- Run to termination
- Step over a node
- Step into or out of a subflow
- Step over, into, or out of, source code

### Breakpoints view

Lists the breakpoints that have been set on connections in your message flow. In this view you can add, disable, enable or remove breakpoints. You can also restrict a breakpoint to one or more specific instances of a message flow using the Properties view.

The Breakpoints view and the Variables view share the same pane. Click one of the tabs to select the view that you want.

The Breakpoints view displays the breakpoints that are set in all instances of a selected flow. Each breakpoint is identified by one of two symbols (as also used in the Message Flow editor) as follows:

-  breakpoint enabled
-  breakpoint disabled

In the Breakpoints view, you can perform the following debugging tasks:

- Remove breakpoints
- Disable or enable breakpoints
- Restrict breakpoints to one or more instances of a flow

### Variables view

When a message flow is interrupted by a breakpoint, you can view the message content to check whether the message flow is executing as expected, and to make any changes required.




The Variables view and the Breakpoints view share the same pane. Click one of the tabs to select the view that you want.

The Variables view displays the messages that are currently traveling through the flow. Use the view to examine or change the content of a message in a flow during debugging.

### Message Flow editor view

The Message Flow editor graphically displays and debugs flows. In this view you can add breakpoints to the connections of a message flow.

The Message Flow editor also displays any breakpoints that are set in the flow. Each breakpoint is identified by a symbol as follows:

-  breakpoint enabled.
-  breakpoint disabled.
-  flow paused at breakpoint.

Also the editor displays the following symbol above a node:

-  flow paused at a node containing ESQL code or Java code that the flow debugger can step into

In the Broker Application Development perspective, the Message Flow editor is used to create, graphically display, and edit flows. For details of

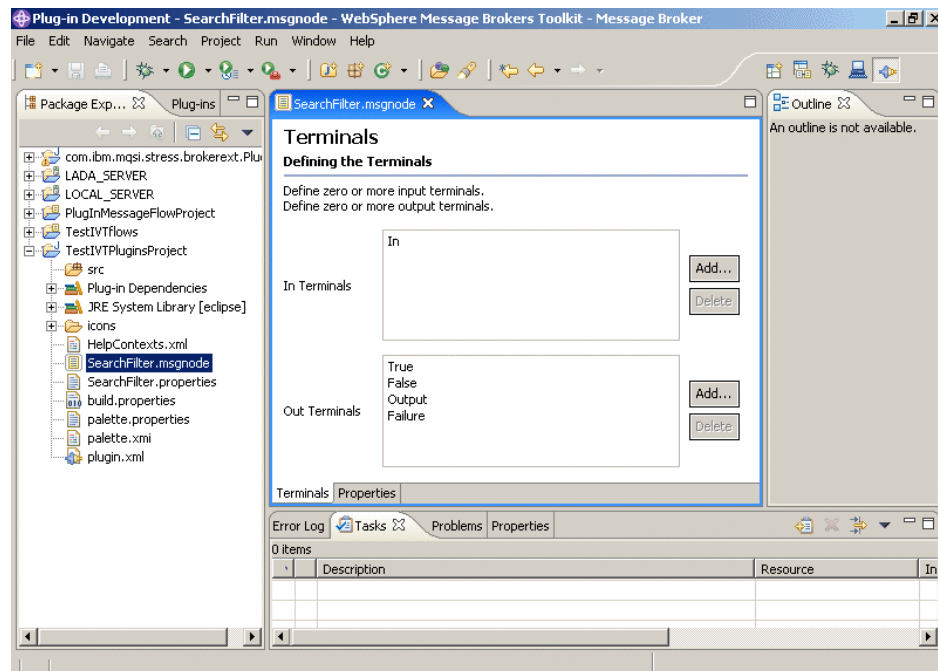
the other uses of this editor, see the description in “Message Flow editor” on page 102 and the tasks in Defining message flow content.

## Plug-in Development perspective

This perspective helps you develop user-defined nodes in the Message Brokers Toolkit. A user-defined node is an Eclipse plug-in that adds a category of nodes to the Message Flow editor palette.

The New Plug-in Project wizard creates the supporting workbench files for your user-defined node. The New Message Flow Plug-in Node wizard creates a message node file (.msgnode) and launches the Message Node editor for developing the visual representation of your user-defined node in the Message Brokers Toolkit.

The following figure shows the Plug-in Development perspective in the Message Brokers Toolkit.



---

## Editors in the Message Brokers Toolkit

Follow the links below for more information about each of the editors provided by Message Brokers Toolkit:

- “Broker Archive editor” on page 91
- “Broker Topology editor” on page 93
- “ESQL editor” on page 93
- “Event Log editor” on page 96
- “Message Category editor” on page 97
- “Message Definition editor” on page 98
- “Message Flow editor” on page 102
- “Message Mapping editor” on page 107
- “Message Node editor” on page 108

- “Message set editor” on page 109
- “Subscriptions Query editor” on page 110
- “Topics Hierarchy editor” on page 111

## Editor preferences and localized settings

If users within the broker domain want to share the files that are associated with message flow and message set development, you must ensure that the files and the content of these files is compatible and can be shared by users working on different systems and in different locales.

Users can share the following files:

- Message flow definition files (.msgflow)
- ESQL files (.esql)
- Mappings files (.mfmap)
- Message set files (.mset)
- Message files (.mxsd and .xsd)
- Message category files (.category)

You are recommended to restrict the names of these files to a specific set that all your users can understand, for example the characters in the US English code page (a to z, A to Z, 0 to 9, plus special characters underscore and hyphen). This is not a restriction, nor is it enforced, but the users must be able to use each others resources without confusion or error.

The contents of all files listed above, with the exception of ESQL files, are stored in UTF-8 and are therefore common regardless of individual user settings.

The contents of the ESQL files, which users can edit directly, are governed by the editor preference settings that you specify in the workbench:

1. Click **Window** → **Preferences**.
2. Expand the Workbench item on the left. Click **Editors**. Within “Text file encoding”, select the code page that you want content to be stored in:
  - a. If you select Default, this represents the default code page for the locale in which you are currently working.
  - b. If you select Other, you can choose one from the list of available code pages.

If you expect users to share ESQL files, either through the use of shared drives across a LAN, or through a shared repository such as CVS, you must ensure that all users set the editor preference to be the same value. This allows each user to work in their current locale, but to be able to access and work with files created by any other user. You are recommended to select UTF-8 to ensure consistent ESQL file content that is accessible by all users regardless of current locale.

## Broker Archive editor

The Broker Archive editor is the component that you use to edit and save bar files. Using this editor, you can configure system object names contained within a bar file without recompiling a message flow. You can also use command line tools to create bar files.

The Broker Archive editor is available in the Broker Application Development perspective and has two sections; one that allows you to add or remove message

flows and message sets to or from your broker archive, and one that lists the configurable properties within your bar file, and allows you to update them.

The Broker Archive editor allows you to add only message flows that are syntactically complete, that is, it does not allow you to add any flows that do not contain an input node, or that contain errors such as mandatory properties that are not set correctly.

## Broker Archive editor views

The Broker Archive editor includes these views:


### Content view

The Content view displays the files that are currently in the broker archive file.

### Configure view





The Configure view displays the configurable properties of your broker archive file deployment descriptor. To edit a property, replace the current value with a new value.

### User Log and Service Log views

The two log views show the contents of the user and service logs. You can copy the text, but you cannot modify it. Click **Remove** () in these views to clear the logs.

## Broker Archive editor toolbar

The icons in the toolbar and their actions are shown in the table.

Icon	Label	Action
	<b>Add</b>	To add message flows or message sets to this archive.
	<b>Remove</b>	To remove message flows or message sets from this archive.
	<b>Edit</b>	To edit selected message flows or message sets in this archive.
	<b>Refresh</b>	To refresh this archive.

A column in the bar editor called Version displays the version tag for all objects that have a defined version. These are:

- .dictionary files
- .cmf files
- Embedded JAR files with a version defined in a META-INF/keywords.txt file

You cannot edit the Version column.

You can use the mqsireadbar command to list the keywords that are defined for each deployable file within a deployable archive file.

## Broker Topology editor

The Broker Topology editor is the default editor that is provided by the Broker Administration perspective for managing brokers and collectives, and for saving topology changes to the Configuration Manager in a single step.

The Broker Topology editor is launched in the editor area of the Broker Administration perspective when you double-click Broker Topology in the Domains view.

If you are not connected to a domain, a connection attempt is made before opening the editor. During this attempt, if you click **Cancel**, the connection that is in progress stops and the domain returns to its initial unconnected state.

When the editor is launched, an overview pane is displayed in the Outline view. The Outline view is stacked on top of the Properties view. Use the **Properties** tab or the **Outline** tab to select which of these two views is displayed.

Use the Broker Topology editor to do the following tasks:

- Create brokers, rename brokers, delete brokers, or change broker properties.  
In the workbench, a broker is a representation of a broker that exists in the domain. When you create or rename a broker in the Broker Topology editor, you must use the name of the physical broker that exists in the domain.  
When you create a broker in the workbench, a red cross at the top right of the broker icon indicates that the **Queue Manager Name** property has not been entered. Right-click the broker, select **Properties** from the shortcut menu, and then enter the name of the queue manager that the broker uses. You cannot save the topology until you have specified a queue manager for each broker.
- Connect brokers to collectives, remove connections between brokers and collectives, or delete collectives.  
The rules for creating a broker topology are given in the Broker Networks section of Publish/subscribe topologies.

When you save topology changes to the Configuration Manager, the topology information in the Domains view on every workbench that is connected to the Configuration Manager is refreshed automatically.

For information about the palette, and how to customize its layout and settings, see Changing the palette layout, Changing the palette settings, and Customizing the palette.

By default, the Broker Topology editor has a background image of a world map. You can change the background image, and modify its scale factor, or choose not to display an image, by following the instructions in Changing Broker Topology editor properties.

## ESQL editor

The ESQL editor is the default editor provided by the Broker Application Development perspective for editing ESQL (.esql) files.

The editor is launched in the editor area when you select the menu item **Open ESQL** for a Compute, Database, or Filter node, or when you double click an ESQL file in the Broker Development view.

## ESQL editor views

The ESQL editor has the following views:

### Broker Development view

The Broker Development view shows all the resources in your workspace, that is all message set resources and all message flow resources, including ESQL files.

### Editor view

The editor view shows the contents of the resource that is currently open. It also shows tabs for each of the resource that you have open so that you can quickly switch between them.

### Outline view

The Outline view displays any schemas, defined constants, modules, and routines that you have referenced in this ESQL file.

### Problems view

The problems view displays the warning and error messages that are generated by the editor's validation when you save the ESQL file. If you double click an error, the editor indicates where it is located by moving the cursor to the corresponding ESQL code.

## ESQL editor functions

The ESQL editor provides:

- A context sensitive Content Assist. You can invoke Content Assist from the Edit menu or, on some systems, by pressing Ctrl+Space if that has not been assigned to another function.

Content Assist helps you construct references to the content of the Properties folder. When you use the ESQL editor with predefined messages, it also helps you construct field references.

When you use the ESQL editor with the database schema definitions, Content Assist helps you construct schema, table, and column references. You can also use the ESQL editor to call user-defined maps.

When you create functions and procedures within the ESQL file, the names that you define must not start with the characters IBM\_ (IBM underscore).

**Note:** To get Content Assist to work, you must set up a project reference from the project containing the ESQL or mappings to the project containing the message set. For information about setting up a project reference, see "Project references" on page 66.

- Automatic code formatting.

Right-click in the editor view to access the following additional functions:

- **Undo** and **Revert File**. To undo a change that you have made to the ESQL file, click **Undo**. If you want to undo all of the changes since the last time you saved you can reinstate them by clicking **Revert File**.
- **Cut**, **Copy**, and **Paste**. These are standard editor functions.
- **Shift Right** and **Shift Left**. These are standard editor functions.
- **Save**. Click this to save your changes.
- **Comment** and **Uncomment**. Click **Comment** to change a line of ESQL code into a comment. Click **Uncomment** to change a comment into a line of ESQL code.

- **Format.** This function formats all selected lines of code (unless only partially selected, when they are ignored), or, if no lines are selected, formats the entire file (correcting alignments and indentation).
- **Organize Schema Paths and Add Schema Path.** These functions assist you with broker schema management.

Click **Organize Schema Paths** and any broker schema containing procedures or function called by the ESQL file is automatically added to the PATH statement (if you have not already added it). This function scans the ESQL file for instances of procedures or function residing in schemas not already fully qualified in the file.

Click **Add Schema Path** when you code a call to a procedure or function residing in a different broker schema to any you have included on the PATH statement, and this schema is added to the PATH statement. Ensure that the cursor is on the name of the procedure you are calling.

## ESQL editor preferences

You can modify settings that affect the way the ESQL code is handled:

- Editor Settings (how code is displayed in the editor view):
  - Text font
  - Displayed tab width (default 4)
  - Background and foreground colors (for comments, statements, and so on)
- Validation Settings (what level of validation is performed when you save the file):

Validation detects four potential problems:

### Unresolved identifiers

The validator attempts to resolve any identifiers that you have referenced (for example, a message field).

### Message references mismatch message definition

If a message definition exists (messages in the MRM domain only), the validator checks that the use of the reference is consistent with its definition (for example, the action against a numeric field is a valid numeric action).

### Database references mismatch database schema

The validator checks that the use of the reference is consistent with the database schema (for example, the action against a numeric field is a valid numeric action).

### Use of deprecated keywords

The validator checks if you have used any keywords that have been deprecated in this release.

For each of these situations, select one of the following validation settings:

**Ignore** No validation is performed.

### Warning

The validator writes warning messages to the Tasks view for each potential problem that it detects. This is the default setting.

**Error** The validator writes error messages to the Tasks view for each potential problem that it detects.

Validation does not check that you have specified names in the case in which you declared them. The names of modules, functions, and procedures are not case sensitive; all other names (schemas, constants, variables, and labels) are case



sensitive. Check that the names that you use match the declarations for those names because the broker handles these names in a case sensitive way and generates a runtime error if they do not match.

For details of how to change these preferences, see [Changing ESQL preferences](#).

## ESQL editor toolbar

The ESQL editor does not provide any additional icons and actions on the toolbar.

## Event Log editor

The Event Log editor is launched in the editor area of the Broker Administration perspective when you double-click the Event Log in the Domains view.

If you are not connected to a domain, an attempt to do so is made before opening the editor. During this attempt, if you click **Cancel**, the connection that is in progress stops and the domain returns to its initial unconnected state.

The event messages are stored and managed by the Configuration Manager. You can view the messages in the Event Log editor following configuration deploys.

The Event Log editor has two panes called Logs and Details. You can maximize and minimize each pane, and toggle between them.

### Event Log editor views

#### Logs pane

The Logs pane displays a summary of each event in the Event Log editor, comprising a message identifier, the source of the message, for example the Configuration Manager or a broker, and the date and time the message was generated.

Right-clicking in the Logs pane displays the following menu items:

- **Clear Log**

Clears all log entries from the Event Log editor and deletes the log entries from the configuration repository. You cannot retrieve a message that has been cleared.

Messages are automatically cleared after 72 hours.
- **Filter Log**

Displays the Log Filter. You can filter on the type, the source and the timestamp of the message to restrict the number of log entries that are displayed in the Event Log editor. You can also mask events, event by event, to make the other entries easier to view in the Event log editor. The filter settings you define are kept for your next session.
- **Revert**

Refreshes the Event Log editor with the most recent log entries from the Configuration Manager. You only need to use this option if the workbench is not synchronized with the Configuration Manager.
- **Save Log As**

Saves the messages to a filename and path of your choice. **Save Log As** does not remove messages from the Event Log editor or the configuration repository.



## Details pane

The Details pane displays the complete message, and any corrective action, for the log entry selected in the Logs pane.

## Preferences

You can change the preferences for the Event Log editor. For example, you can customize the colors used for each type of event message.

## Message Category editor

The Message Category editor is the default editor provided by the Broker Application Development perspective for working with message category (.category) files in a message set.

The editor is launched in the editor area when you open an existing message category file using the Broker Development view, or when you create a new message category file using the New Message Category File wizard.

### Message Category editor views

The Message Category editor has the following views:

#### Broker Development view

The Broker Development view shows a hierarchical view of all the resources that are currently in your workspace. By expanding the folder for a message set project, you can see the resources, including the message category file or files, that this message set project contains.

#### Editor view

You edit the properties of message category in the editor area of the Message Category editor, which has a single tab, the **Properties** tab. This tab provides:

- The Properties Hierarchy, which shows the hierarchy for a message category file and the messages that have been added to it.
- The Details view, which when you select a message category file, shows the properties of the message category, or of an individual message in the category, depending on what you have selected in the Properties Hierarchy.

There are tabs across the top of the editor area for each file that you have open, so that you can quickly switch between them.

#### Tasks view

Each time you save a change within a message set project, the content is validated to ensure that the message model follows certain rules. Any informational, warning or error messages relating to the validation appear in the task list.

### Properties Hierarchy functions

Right-click in the Properties Hierarchy to display the following menu items:

- **Add Messages** (available at the **Message Category** level only) adds a new message to a message category file.
- **Undo** and **Redo** allow you to undo and redo changes to message category files. These two menu items apply to adding new messages, and also to changes that

you make in the Details view. To undo a change that you have made to a message category file but not yet saved, click **Undo**. If you undo a change, you can reinstate it by clicking **Redo**.

- **Delete** (available at the individual **Message** level only) deletes the selected message from the message category file.

## Details view functions

In the Details view, you use the **Message Category Kind** field to specify whether the message category will be used to generate WSDL (Web Services Description Language) files.

If the **Message Category Kind** is WSDL, you must provide definitions in the **Role Name** and **Role Type** fields for each message that you add to the message category file. These two fields appear when you select a message in the Properties Hierarchy.

You create any required documentation for a message set in the **Documentation** field. Right-click in this field to display a menu of standard text editing menu items that apply specifically to this field and allow you to undo changes, cut, copy, paste and delete text, and select all the text in the field.

## Message Definition editor

The Message Definition editor is the default editor provided by the Broker Application Development perspective for editing message definition (.mxsd) files.

The editor is launched in the editor area when you open an existing message definition file using the Broker Development view, or when you create a new message definition file using the New Message Definition File wizard.

You use the Message Definition editor to:

- Edit message definitions created by importing data structures using the XML Schema, DTD, C or COBOL importers. The message definition file that the import process creates is automatically populated with the imported content, which you can then edit as required.
- Populate empty message definition files with message model objects by creating the elements, attributes, groups, types and messages needed to represent your message formats. The message model that you create can consist of both logical and physical information, if appropriate physical formats exist in the message set.

## Message Definition editor views

The Message Definition editor has the following views:

### Broker Development view

The Broker Development view shows a hierarchical view of all the resources that are currently in your workspace. By expanding the folder for a message set project, you can see the resources, including message definition file or files, that this message set project contains.

### Outline view

You use the Outline view to select message set objects to display in the editor area. When you open a message definition file, the Outline view

shows a hierarchical view of the messages, types, groups and elements and attributes that the selected message definition contains.

### Editor view

You edit the properties of message set objects in the editor area, which displays the message definition information for the object that you have currently selected in the Outline view. There are tabs across the top of the editor area for each file that you have open, so that you can quickly switch between them.

In the editor area, you can make any of the following types of change:

- Edit the logical structure of a message.
- Create and edit the physical structure and properties of a message.
- Create message definitions.
- Create common constructs within a message set for use with other message definition files.

The Message Definition editor comprises an Overview editor and a Properties editor. You switch between them by clicking the two tabs in the bottom left of the editor area. You can change the order in which the tabs are displayed by changing the message set preferences.

### Tasks view

Each time you save a change within a message set project, the content is validated to ensure that the message model follows certain rules. Any informational, warning or error messages relating to the validation appear in the task list. Double clicking a message displays the properties of the relevant object in the editor area and shows the location of this object by highlighting the appropriate levels in the Broker Development view and the Outline view.

## Message Definition editor functions

For information on the functions that the Message Definition editor provides, refer to the following topics:


- “Message Definition editor: Outline view”
- “Message Definition editor: Overview editor” on page 100
- “Message Definition editor: Properties editor” on page 101



### Message Definition editor: Outline view


When you open a message definition file, the hierarchy for the selected file appears in the Outline view. You use the Outline view to select objects so that you can view and edit their message definition details in the editor area.

#### Global actions:

The Outline view toolbar icons provide a number of global actions that apply to all levels of the displayed hierarchy. The icons in the toolbar and their actions are shown in the table below.

Icon	Label	Action
	Find Element	Finds the specified message definition construct.

Icon	Label	Action
	<b>Sort global constructs</b>	If selected, global constructs are sorted alphabetically. If not selected, global constructs are displayed in the order that you create them.
	<b>Collapse All</b>	Collapses all the levels that are currently displayed in the Outline view.

You can also select these global actions from the toolbar drop-down menu, which appears when you click the  icon.

### Common editing and navigation actions:

The Outline view provides the following common actions, which are also available in the Overview editor. Right-click in the Outline view or Overview editor to display the menu items.

- **Copy** and **Paste** provide standard editing functions.
- **Undo** and **Redo** allow you to undo and then redo changes that you have made. To undo a change, click **Undo**. If you undo a change, you can reinstate it, if you want to, by clicking **Redo**.
- **Go To Declaration** takes you to the appropriate point in the hierarchy (global group from a group reference, global attribute from an attribute reference, global element from an element reference, global element from a message) either in the same or in a different message definition file.
- **Go to Type Definition** takes you to the appropriate point in the hierarchy.
- **Delete** deletes the selected message set object.

### Options for adding message model objects:

The Outline view provides options for adding the message model objects listed below. These options are also available in the Overview editor.

- Messages
- Complex types
- Simple types
- Attribute groups
- Groups
- Elements
- Attributes

Right click the appropriate level of the displayed hierarchy to display a pop-up menu with the menu items available for the selected level.

### Message Definition editor: Overview editor

The Overview editor displays the contents of the message set object that you select in the Outline view so that you can view and change this object's most common properties. The Overview editor appears when you click the Overview tab in the editor area. The navigation buttons and links on the toolbar allow you to move back and forth between different objects. In the Overview editor you can make the types of change listed below:

- Edit the displayed values of any of the following:

- Name
- Type
- Min Occurs
- Max Occurs

To edit any of these values, select then click the appropriate field in the Overview editor. Depending on the type of field, this either displays a drop-down list or enables the field for direct editing.

- Add message model objects as in the Outline view. To add a new message model object, right-click **Structure** then click the appropriate menu item from the displayed pop-up menu. For example, the menu for **Messages** provides **Add Message** and **Add Message from Global Element**. When you add a new message model object, this appears in both the Outline view and the Overview editor.

### Message Definition editor: Properties editor

The Properties editor allows you to change the properties for the selected message set object. The information that appears depends on what is currently selected in the Outline view and Overview editor. The Properties editor appears when you click the Properties tab in the editor area. The navigation buttons and links on the toolbar allow you to move back and forth between different objects. The Properties editor comprises the Properties Hierarchy and Details view.

#### Properties Hierarchy functions:

When you select a object in the Outline view, the Properties Hierarchy shows the following hierarchy for the message definition file that is currently open in the Message Definition editor:

- Logical properties
- Physical properties
- Documentation

Right-click in the Properties Hierarchy to display a number of menu items. The displayed menu items depend on which level of the hierarchy you select:

- **Undo** and **Redo** are available for all levels in the Properties Hierarchy and apply to changes that you make in the Details view for the currently selected object. To undo a change that you have made to a file but not yet saved, click **Undo**. If you undo a change, you can reinstate it by clicking **Redo**.
- **Add Include** and **Add Import** are available when the top level of the message definition (.xsd) file's hierarchy is displayed in the **Outline** view. They allow you to link message definition files together.
- **Apply default physical format settings** is available for **Physical formats** only and applies the default wire format settings for the selected object.

If you add a new physical format in the message set (messageSet.mset) file, this new format is added under **Physical Properties** in the Properties Hierarchy. When you click **Physical Properties**, a link to the Message Set editor appears in the Details view.

#### Details view functions:

The Details view shows the properties for the object that is currently selected in the Properties Hierarchy. You can save any changes to the properties as you make them.

You create any required documentation for a message set in the **Documentation** field. This field appears when you select an object under **Documentation** in the Properties Hierarchy. Right-click in this field to display a menu of standard text editing menu items that apply specifically to this field and allow you to undo changes, cut, copy, paste and delete text, and select all the text in the field.

## Message Flow editor

The Message Flow editor is the default editor provided by the Broker Application Development perspective for defining a graphical representation of a message flow in the workbench, and for setting properties for individual message flow nodes.

Open a message flow file (.msgflow) in the Broker Development view to launch the appropriate editor in the Editor view. The Editor view is where you select built-in and user-defined nodes, and the connections between them, to define a message flow.

### Message Flow editor views

The Message Flow editor has the following views:

#### Broker Development view

The Broker Development view shows all the resources in your workspace; that is, all message set resources and all message flow resources.

#### Editor view

The editor view shows the contents of the resource that is currently open. It also shows tabs for each of the resources that you have open so that you can quickly switch between them.

When the resource that is open is a message flow, at the bottom left of the editor view, there are two tabs, **Graph** and **User Defined Properties**.

When you select **Graph** (the default), the Editor view contains a graphical display of the message flow.

To the left of the editor view, there is a palette bar that contains all the available nodes that you can include in the message flow. By default, the palette bar is shown in collapsed mode when you open a message flow.

To open the palette, use one of the following methods:

- Hover the mouse over the palette bar while it is in collapsed mode. The palette bar expands. When you move the mouse away from the palette bar, it collapses again.
- Click the **Show Palette** icon at the top of the palette bar. The palette bar expands and it remains expanded when the mouse is moved away from the palette bar. To collapse the palette bar again, click the **Hide Palette** icon at the top of the palette bar while it is in expanded mode.

By default, the palette appears on the left side of the editor view. You can move it to the right side of the editor view by clicking on the Palette bar while it is in collapsed mode, and dragging the bar to the right-side of the editor view.

When you select the **User Defined Properties** tab, a User Defined Properties editor is opened that allows you to change the User Defined Properties of the message flow. The User Defined Properties editor consists of a User Property Hierarchy view and a Details view.

The User Property Hierarchy view displays three icons, **Add Property Group**, **Add Property**, and **Delete** that you can use to update the property hierarchy. When you add a property, a Details view is opened. In the Details view you can define the property Type and Default Value.

### Outline view

The Outline view enables you to navigate to a particular node in a message flow, and edit its properties.

### Palette view

The Palette view lists the available nodes that you can select and include in the message flow. For more information about the palette and how to customize the layout and settings, see Message flow node palette.

### Overview view






The Overview view provides a useful summary for large, complex message flows because it shows a small-scale version of the flow. Click the tab to show or hide the outline view.








### Tasks view

The tasks view displays the warning and error messages that are generated by the editor's validation when you save the message flow file. If you double-click an error, the editor indicates where it is located (for example, if you have not set a mandatory property in a node, it opens the Properties view for that node on the right page).

## Message Flow editor toolbar

The icons in the toolbar and their actions are shown in the table below.

Icon	Label	Action
	Manhattan	Displays all node connections as a series of horizontal and vertical lines
	Show grid	Displays a grid of horizontal and vertical dotted lines in the background of the editor area.
	Grid properties	Defines the horizontal and vertical spacing of the grid markers, and the gap between the borders of the editor area and the start of the grid markers.
	Align left	Lines up the left edge of the currently selected nodes. Enabled only when more than one node is selected.
	Align center	Lines up the horizontal center point (between left and right) of the currently selected nodes. Enabled only when more than one node is selected.

Icon	Label	Action
	Align right	Lines up the right edge of all the currently selected nodes. Enabled only when more than one node is selected.
	Align top	Lines up the top edge of the currently selected nodes. Enabled only when more than one node is selected.
	Align middle	Lines up the vertical center point (between top and bottom) of the currently selected nodes. Enabled only when more than one node is selected.
	Align bottom	Lines up the bottom edge of the currently selected nodes. Enabled only when more than one node is selected.
	Show distribute box	Displays a rectangular box around the currently selected nodes.
	Distribute horizontally	Aligns the currently selected nodes with the nearest right or left edge within the distribute box.
	Distribute vertically	Aligns the currently selected nodes with the nearest top or bottom edge within the distribute box.

## Message Flow editor hover-feedback

In the Message Flow editor, you can display node and connection metadata by hovering the mouse over a node or subflow in a message flow. To view metadata information for a node, subflow, or connection:

1. Switch to the Broker Application Development perspective.
2. Open a message flow.
3. In the Message Flow editor, hover the mouse over a node, a subflow, or a node connection in the open message flow by placing the mouse over the element.

A custom tooltip is displayed below the element.

- To turn the pop-up window into a scrollable window, press **F2**.
- To hide the pop-up window, either press **Esc** or move the mouse away from the node.

## Node and subflow feedback

When you hover the mouse over a node or subflow in the Message Flow editor, the pop-up window contains the following metadata information:

- The first line of text contains the node type and the display name of the node instance. This line appears as **bold text**.



- The line below the node type and display name contains the node's short description.
- The final line contains the node's long description.

If the node or subflow has no short or long description associated with it, only the first line of text is displayed.

If the task list contains errors or warnings that are associated with the node or subflow, the error or warning information is displayed instead of the metadata information.

### Connection feedback

When you hover the mouse over a connection in the Message Flow editor, hover-help is displayed, containing the names of the output and input terminals that are connected.

### Message Flow editor menus

All the actions that you can perform in the Editor view of the Message Flow editor are also available from a series of drop-down menus in WebSphere Message Broker. When the Message Flow editor is open, the following menus appear in the Workbench:

Menu	Keyboard shortcut
Flow	Alt + O
View	Alt + V
Palette	Alt + L
Run	Alt + R

The Navigate menu also contains actions specific to the Message Flow editor.

### Flow menu

The flow menu contains all the actions that address editing of the flow model. Actions are statically positioned on the menu, and are enabled or disabled when nodes or connections are selected in the editor.

The following actions are available:

Menu item	Keyboard shortcut
Create Connection	C
Add subflow...	A
Rename...	N
Locate Subflow...	L
Promote Property	P
Properties...	R

## View menu

The View menu contains all actions specific to the visual presentation of the flow model.

Menu item	Keyboard shortcut
Zoom In	I
Zoom Out	O
Manhattan Layout	M
Show Grid	G
Snap to Grid	G
Grid Properties...	P
Align	A
Distribute	D
Layout	L
Rotate	R

## Palette menu

The Palette menu contains all actions specific to the Message Flow editor Palette.

Menu item	Keyboard shortcut
Add Node to Canvas	N
Customize	T
Settings...	S
Revert Palette to Defaults	T

## Run menu

The Run menu contains the flow debug actions.

Menu item	Keyboard shortcut
Add Breakpoint	A
Remove Breakpoint	E
Add Breakpoints After Node	A
Add Breakpoints Before Node	B

## Navigate menu

The Navigate menu contains the following actions specific to the Message Flow editor:

Menu item	Keyboard shortcut
Open Subflow	O
Open Source	S

## Adding a node using the keyboard

### Before you start:

- Create a message flow or open an existing message flow
- Read the concept topic about message flow nodes

You can use the keyboard to perform tasks in the Message Flow editor, such as adding a node to the canvas.

1. Switch to the Broker Application Development perspective.
2. Open the message flow to which you want to add a node.
3. Open the Palette view or the Palette bar.
4. Select a node in the Palette view or Palette bar using the up and down arrows to highlight the node that you want to add to the canvas.
5. Add the Node to the canvas using one of the following methods:
  - Press **Alt + L**, then press **N**.
  - Press **Shift + F10** to open the context-sensitive menu for the Palette, and press **N**.

The node that you selected in the Palette bar or Palette view is placed on the canvas in the Editor view.

When you add a node to the canvas, the editor automatically assigns a name to the node, but the name is highlighted and you can change it by entering a name of your choice. If you do not change the default name at this time, you can change it later. The default name is set to the type of node for the first instance. For example, if you add an MQInput node to the canvas, it is given the name MQInput; if you add a second MQInput node, the default name is MQInput1; the third is MQInput2, and so on.

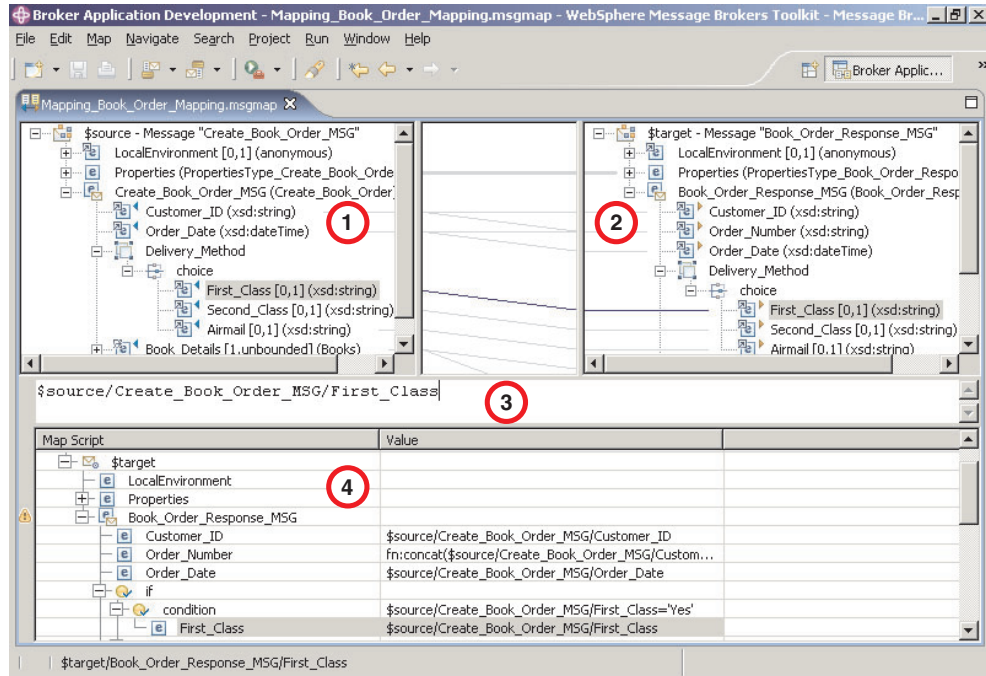
You can move the node that you have placed on the canvas using the keyboard controls described in “Message Brokers Toolkit keyboard shortcuts” on page 115.

## Message Mapping editor

You configure a message mapping using the Message Mapping editor, which you use to set values for:

- the message destination
- message headers
- message content

Here is an example of the Message Mapping editor. There are separate panes for working with sources, targets and expressions, as well as a spreadsheet view.



1. **Source pane:** displays a source message or database table
2. **Target pane:** displays a target message
3. **Edit pane:** displays the expression to be used to derive the target element value
4. **Spreadsheet pane:** displays a summary of the mappings in spreadsheet columns (each target field and its value)

Use the Message Mapping editor to perform various mapping tasks.

Wizards and dialog boxes are provided for tasks such as adding mappable elements, working with ESQL, and working with submaps. Mappings that are created with the Message Mapping editor are automatically validated and compiled, ready for adding to a broker archive (.bar) file, and subsequent deployment to WebSphere Message Broker.

## Message Node editor

The Message Node editor is the default editor provided by the Plug-in Development perspective for defining the content of user-defined nodes.

The editor is launched in the editor area of the perspective when a new message flow node file (.msgnode) is created.

### Message Node editor views

The Message Node editor has two tabs:

#### Terminals

This tab is for adding, renaming, or deleting input and output terminals for the new node.

#### Properties

This tab is for adding properties and property groups to the node, and defining property type.

## Message set editor

The Message set editor is the default editor provided by the Broker Application Development perspective for editing message set (messageSet.mset) files.

The editor is launched in the editor area when you open an existing message set file using the Broker Development view, or when you create a new message set file using the New Message Set File wizard.

### Message set editor views

The Message set editor has the following views:

#### Broker Development view

The Broker Development view shows a hierarchical view of all the resources that are currently in your workspace. By expanding the folder for a message set project, you can see the resources, including the message set file, that this message set project contains.

#### Editor view

You edit the properties of message set in the editor area of the Message set editor, which has a single tab, the **Properties** tab. This tab provides:

- The Properties Hierarchy, which provides a hierarchy with four basic categories: **Message Set**, **XML Wire Formats**, **Custom Wire Formats**, and **Tagged/Delimited String Formats**.
- The Details view, which displays the properties relating to the currently selected level in the Properties Hierarchy.

There are tabs across the top of the editor area for each file that you have open; this enables you to quickly switch between the open files.

#### Tasks view

Each time that you save a change within a message set project, the content is validated to ensure that the message model follows certain rules. Any informational, warning, or error, messages that relate to the validation appear in the task list.

### Properties Hierarchy functions

Right-click in the Properties Hierarchy to display the menu items available for the level of the hierarchy that you have currently selected:

- **Undo** and **Redo** are available for all levels in the Properties Hierarchy and apply to changes that you make in the Details view. To undo a change that you have made to a message set file, but have not yet saved, click **Undo**. If you undo a change, you can reinstate it by clicking **Redo**.
- If **XML Wire Formats**, **Custom Wire Formats**, or **Tagged/Delimited String Formats** are selected, menu items are available for adding physical format layers to a message set file. These are **Add XML Wire Format**, **Add Custom Wire Format**, and **Add Tagged/Delimited String Format**, respectively.
- If an existing physical format is selected, the following menu items are available for the selected physical format layer:
  - **Apply default physical format settings** applies the default physical format settings to the selected physical format layer.
  - **Rename** renames the selected physical format layer for the message set file. Click **Finish** on the displayed window to confirm the change of name.

- **Delete** deletes the selected physical format layer from the message set file. Click **Finish** on the displayed window to confirm the deletion.

## Details view functions

You use the Details view to define and edit global properties for the message set, and to create and edit properties for the physical format layers that you have added to the message set.

You create any required documentation for a message set in the **Documentation** field. Right-click in this field to display a menu of standard text editing menu items that apply specifically to this field and that allow you to undo changes, cut, copy, paste and delete text, and select all the text in the field.

## Subscriptions Query editor

The Subscriptions Query editor is the default editor that is provided by the Broker Administration perspective for managing subscriptions. Use this editor to view and delete subscriptions in your publish/subscribe system. The editor allows you to set query parameters, submit subscription queries, and delete subscriptions.

The Subscriptions Query editor is launched in the editor area of the Broker Administration perspective when you double-click Subscriptions in the Domains view.

If you are not connected to a domain, the workbench tries to make the connection before it opens the editor. If you click **Cancel** during this attempt, the connection that is in progress stops, and the domain returns to its initial unconnected state.

### Subscription Query editor view

The Subscription Query editor view is divided into two parts. In the upper part of the edit window, you can define your query. When you click **Query**, the current subscriptions are displayed in the lower part of the edit window. Unless you make any changes to the default query, the edit window displays all subscriptions that are registered with all of the brokers that are set up for publish/subscribe.

### Subscription query

You can change the default query by changing the values that are displayed in any of the entry fields in the upper part of the edit window. You can use wildcard characters in the entry fields to display a specified range of subscriptions. The wildcard characters are:

A percent character `'%'` for zero or more characters. For example, `'B%EAD'` would match character strings where zero or more characters follow B and precede the string EAD, like BREAD, BEAD, or BROWNBREAD.

An underscore character `'_'` for a single character. For example, `'_ATCH'` would match five-character strings where any single character precedes the string ATCH, like BATCH, HATCH, or LATCH.

The entry fields are:

- **Topics**

If you don't want to display subscriptions for all topics, you can put a topic name here. You can use the wildcard characters to display subscriptions for a range of topics.

- **Brokers**

If you want to display only those subscriptions that are valid for a specific broker, you can put the name of the broker here. You can use the wildcard characters to display the subscriptions for a range of broker names.

- **Users**

If you want to display only those subscriptions that are valid for a specific user, you can put the user name here. You can use the wildcard characters to display the subscriptions for a range of user names.

- **Subscription points**

If you want to display only those subscriptions that are valid for a specific subscription point, you can put the name of the subscription point here. You can use the wildcard characters to display the subscriptions for a range of subscription points.

You can also specify that you want to display subscriptions that were registered between two dates.

### **Result of a query**

The result of a query is displayed in a table in the lower part of the edit window. Each row of the table describes a registered subscription request.

The table has columns that contain the following information:

- The name of the topic
- The name of the user that registered the subscription
- The name of the broker where the subscription was registered
- The subscription point (if any) that is associated with the subscription
- The date when the subscription was registered

## **Topics Hierarchy editor**

The Topics Hierarchy editor is the default editor that is provided by the Broker Administration perspective for managing the publish/subscribe topic hierarchy. Use this editor to create, modify, and delete topics, or to define or modify your topic Access Control Lists (ACLs).

The Topics Hierarchy editor is launched in the editor area of the Broker Administration perspective when you double-click Topics in the Domains view.

If you are not connected to a domain, an attempt to do so is made before opening the editor. During this attempt, if you click **Cancel**, the connection that is in progress stops and the domain returns to its initial unconnected state.

In the editor view, use the **Topic/Users** and **User/Topics** buttons to determine whether you see a list of topics or a list of users in the left-hand part of the editor area.

If you click **Topic/Users**, the left-hand part of the editor area contains a list of the topics. Click on a topic to see in the right-hand part of the editor area the ACL for the topic that you selected.

If you click **User/Topics**, the left-hand part of the editor area contains a list of the groups and users. Click on a group or user to see in the right-hand part of the editor area the ACL for the topics that are available to the group or user that you selected.

When the editor is launched, an Outline view is also shown that lists the topics and principals (groups and users).

Either the outline view or the editor view can be used to select what is shown in the right-hand part of the editor area. The behavior varies depending on whether **Topic/Users** or **User/Topics** is selected.

If **Topic/Users** is selected, the following happens:

- If you select a topic in the Outline view, the corresponding topic is selected in the Topics tree.
- If you select a user or group in the Outline view, the corresponding group or user is selected in the Topic ACL table.
- If you select a topic in the Topics tree, the corresponding topic is selected in the Outline view.
- If you select a group or user in the Topic ACL table, the corresponding group or user is selected in the Outline view.

If **User/Topics** is selected, the following happens:

- If you select a topic in the Outline view, the corresponding topic is selected in the User ACL table.
- If you select a user or group in the Outline view, the corresponding group or user is selected in the Users tree.
- If you select a user or group in the Users tree, the corresponding user or group is selected in the Outline view.
- If you select a topic in the User ACL view, the corresponding topic is selected in the Outline view.

The Outline view reacts to changes made in the editor area; for example when a topic is created or removed.

---

## Resource types in the Message Brokers Toolkit

Follow the links below for reference information about the projects and resource files specific to Message Brokers Toolkit:

- “Message flow projects and files”
- “Message set projects and files” on page 113
- “Plug-in Development projects and files” on page 114 (for user-defined nodes and parsers)

### Message flow projects and files

You develop message flows in the Broker Application Development perspective. Message flow resources are maintained within message flow projects. A message flow project compiles to one or more deployable flows.

Message flow projects contain the following resource files:



### **.msgflow files**

A message flow file is a graphical representation of a message flow, containing message nodes and connections. It also contains property values and overrides to define, implement and control the behavior of the message flow nodes. The New Message Flow Wizard automatically creates this file.

Default editor: Message Flow editor

### **.esql file**

(Optional) ESQL modules containing compute, filter, and database ESQL functions and procedures. If none of the message flows defined in the project include any nodes that require ESQL, you do not need to create an ESQL file.

Default editor: ESQL editor

### **.msgmap file**

(Optional) Mapping transformation descriptions for messages and databases. If none of the message flows defined in the project include any nodes that require mappings, you do not need to create a mapping file.

Default editor: Message Mapping editor

You can define more than one message flow, each in a separate `.msgflow` file, in the same message project.

## **Message set projects and files**

A message set project contains all the resources associated with exactly one message set. A message set is a logical grouping of messages and the objects (elements, types and groups) that comprise them. It contains the following resources, which you create and maintain in the Broker Application Development perspective.

### **messageSet.mset file**

A message set file contains those message model properties that are common across all the content of the message set. It also contains the physical format definitions for the message set. A message set project must contain exactly one message set file.

Default editor: "Message set editor" on page 109

### **.mxsd files**

A message definition file contains, in XML Schema form, the logical model and associated physical model for one or more messages. Each message set requires at least one message definition file to describe its properties, and you can have as many as you want within the same message set. One message definition file can, if required, reference XML Schema objects in another message definition file.

You can create a message definition file in a message set by importing any of the following data structures:

- XML Schemas
- XML DTD
- C header files
- COBOL copybooks

Importing any of these data structures automatically creates the message definition file and its content for you. Alternatively, you can create a blank message definition file and add the message definitions yourself.

Default editor: “Message Definition editor” on page 98

#### **Previous versions of the message set model**

The message definition contains the same application data that, previously, was stored in the following separate files:

- Message set (MRProject) that contains wire formats and their default properties
- Messages (MRMessage) that provides a unique name for a message and specifies its element
- Elements (MRElement) that defines a node in the message tree
- Types (MRType) that specify the structure of elements
- Declaration Qualifiers (MRDeclaration Qualifier) that specify constraints applied to elements in the message model

#### **.category files**

A message category file provides you with another way of grouping related messages, for example for documentation generation, or for convenience purposes such as specifying the messages that define a complete request/reply transaction. You can also use a message category file to assist in generating a Web Services Description Language (WSDL) document. Message category files are optional, and you can have as many as you want within the same message set.

Default editor: “Message Category editor” on page 97

Once you have created and populated a message set, you can use it to generate your message model in several different representations for use by your applications:

- A message dictionary for deployment to the WebSphere Message Broker.
- A W3C XML Schema for use by an application that is building or processing XML messages.
- A Web Services Description Language (WSDL) document that specifies the interface for a Web Service.
- An HTML document for use by business analysts and developers.

## **Plug-in Development projects and files**

A plug-in project can contain any Message Brokers Toolkit-developed plug-in resource. In WebSphere Message Broker, it is used for user-defined message nodes.

### **User-defined Message Nodes**

Plug-in development projects used to create Message Nodes contain these resources:

#### **.msgnode file**

Contains the definition of the node.

#### **palette.xmi file**

Lists the user-defined nodes in the project and the Message Editor palette group in which they are shown.

**.html file**

Contains the help files for the node.

**.java file**

Java source code for Property Editors and Compilers.

**plugin.xml file**

The plug-in manifest file that adds the node to the WebSphere Message Broker installation

**.properties file**

National Language properties - contains the translations for Properties and Terminals

**.gif file**

Icon - contains the image to associate with the node

## Rules for naming workspace objects

You can uniquely identify and locate a workspace resource using a namespace and a name. An organization uses its own name to form a namespace, and all resource names within that namespace are safe from clashes with those of another organization. The combination of a user specified namespace and name is often called the "fully qualified name". The unnamed namespace (name space="") is supported.

WebSphere Message Broker defines an application symbol space using project references. No fully qualified names can be duplicated across the application symbol space. A Tasks List error is displayed if duplicate or ambiguous names are detected within an application symbol space.

---

## Message Brokers Toolkit keyboard shortcuts

You can navigate the user interface using the keyboard.

The following table shows the general shortcut keys that are provided in the workbench:

Keyboard shortcut	Result
Left, right, up, and down arrow keys	Move in the direction shown on the key.
Tab	Move the focus forward through the workbench.
Shift+Tab	Move the focus backward through the workbench.
Esc	Cancel the action.
Enter	Commit the action.
Spacebar	Select the object that currently has the focus.
Shift+F10	Open a pop-up menu.
Ctrl+F6	Open the Editors window. (Continue to hold down Ctrl to keep the window open. Use the up and down arrow keys to select a file. Release Ctrl. Press Enter.)

## Message Flow editor keyboard shortcuts

The following table shows keyboard shortcuts that are specific to the Message Flow editor:

Keyboard shortcut	Result when the Selection tool is active	Result when the Connection tool is active
Left, right, up, and down arrow keys	Select a node on the palette or on the canvas.  Use with Shift, or Ctrl+Spacebar, to select multiple nodes.	Select a node's terminal.
slash (/) and backslash (\)	Press slash (/) to cycle forwards, or press backslash (\) to cycle backwards through the selected node's connections.  Use with Shift, or Ctrl+Spacebar, to select multiple connections.	
decimal key (.) and Shift+decimal key (.)	Press the decimal key (.) to cycle forwards, or press Shift and the decimal key (.) to cycle backwards through the move handles, or the resize handles, of the currently selected object on the canvas.  When a node is selected, use this shortcut to activate the move handles, then use the arrow keys to move the node. Press Enter to confirm the move.  When a connection is selected, use this shortcut to activate its resize handles, then use the arrow keys to create bend points in the connection. Press Enter to confirm the move.	
Esc		In general, pressing Esc cancels the previous action. When there is no previous action to cancel, and the Connection tool is selected, pressing Esc activates the Selection tool.

The following examples show how to use the Message Flow editor keyboard shortcuts.

**Adding nodes from the palette to the canvas:**

1. Open the palette.
  - a. Open the Editors window.
  - b. Select the file that you want to use.
  - c. Press Tab until the focus moves to the palette arrow.
  - d. Press Enter or Spacebar to give focus to the palette.
2. Press Tab to give focus to the drawers within the palette.
3. Use the up and down arrow keys to highlight the drawer required. Press Enter.
4. Use the up and down arrow keys to highlight the node required.
5. Press Alt+L to open the palette menu, and select **Add Node to Canvas**. Press Enter. The node is now on the canvas.

**Moving nodes on the canvas:**

1. Press Tab to give focus to the canvas.
2. Select a node on the canvas.
3. Press the decimal key (.) to select one of the move handles around the node.

4. Use the left, right, up, and down arrow keys to move the node to a new location.
5. Press Enter to confirm the move.

#### **Connecting nodes:**

1. Press Tab to give focus to the canvas.
2. Select the node on the canvas from which you want to create a connection.
3. Press Shift+F10 to open a pop-up menu.
4. Use the arrow keys to select **Create Connection**. Press Enter. The Terminal Selection window opens.
5. Press Tab until the focus is in the **Select Terminal** field. Use the arrow keys to select a source (output) terminal.
6. Press Tab to select **OK**, press Enter to confirm your selection.
7. Use the arrow keys to select the input terminal of the target node. Press Enter to confirm the connection.

#### **Selecting a node connection:**

1. Press Tab to give focus to the canvas.
2. Use the arrow keys to select a node that has a connection you want to select.
3. Use the slash (/) and backslash (\) keys to cycle through the connections.

#### **Creating a bend point:**

1. Select a node connection.
2. Press the decimal key (.) until the midpoint of the connection is selected.
3. Use the arrow keys to move the midpoint to a new location, and press Enter to confirm the move.

#### **Reconnecting an existing connection to a new terminal:**

1. Select a node connection.
2. Press the decimal key (.) until the end of the connection that you want to modify is selected.
3. Use the arrow keys to select a new source (input) terminal, and press Enter to confirm your selection.



---

## Glossary of terms and abbreviations

This glossary defines WebSphere Message Broker terms and abbreviations that are used in this online information center.

The migration glossary lists differences in terminology between WebSphere Message Broker Version 6.0 and previous versions of the product.

### A

#### **access control list (ACL)**

In computer security, a list associated with an object that identifies all the subjects that can access the object that it is associated with. The list also defines their access rights. Subjects are principals that have explicit permissions (to publish, to subscribe to, and to request persistent delivery of, a publication message) against a topic in the topic tree. The ACLs define the implementation of topic-based security.

**ACL** See access control list.

#### **active working set**

The logical collection of application projects that is currently displayed in the Broker Development view. See also working set.

#### **aggregation**

See message element aggregation.

**AMI** See Application Messaging Interface.

#### **Application Messaging Interface (AMI)**

The programming interface, provided by WebSphere MQ, that defines a high level interface to message queuing services. See also Message Queue Interface (MQI) and Java Message Service (JMS). Applications that use the AMI connect to the broker using WebSphere MQ Enterprise Transport.

#### **attribute**

In XML, a name-value pair within a tagged element that modifies certain features of the element. In other message domains, a simple element within a message. An attribute requires special treatment when used with XML messages. In messages that are not XML, an attribute is treated exactly like a simple element based on the same simple type.

#### **attribute group**

A set of attributes that can appear in a complex type.

### B

#### **bar file**

See broker archive file.

#### **bend point**

A point that is introduced in a connection between two message flow nodes at which the line that represents the connection changes direction. A bend point can be used to make node alignment and processing logic clearer and more effectively displayed.

**binary large object (BLOB)**

A block of bytes of data (for example, the body of a message) that has no discernible meaning, but is treated as one solid entity that cannot be interpreted.

**BLOB** See binary large object.

**BLOB domain**

The message domain that includes all messages that have content that cannot be interpreted or subdivided into smaller sections of information. Messages in this domain are processed by the BLOB parser. See also IDOC domain, JMS domain, MIME domain, MRM domain, XML domain, and XMLNS domain.

**BLOB parser**

A program that interprets a bit stream or message tree that represents a message that belongs to the BLOB domain, and generates the corresponding tree from the bit stream on input, or the bit stream from the tree on output.

**broker**

A set of execution processes that host one or more message flows. Also known as message broker.

**broker archive file**

The unit of deployment to the broker; also known as a bar file. It contains any number of compiled message flows (.cmf), message sets (.dictionary), and a single deployment descriptor. It can also contain any additional files that you might need, provided that the extension does not overlap the .cmf and .dictionary extensions.

**broker domain**

A collection of brokers that share a common configuration, together with the Configuration Manager that controls them.

**broker schema**

A symbol space that defines the scope of uniqueness of the names of resources that are defined within it. The resources include message flows, ESQL files, and mapping files.

**built-in node**

A message flow node that is supplied by the product. Some of the supplied nodes provide basic processing such as input and output.

**C****callback function**

See implementation function.

**cardinality**

See mapping cardinality.

**category**

An optional grouping of messages that are related in some way. For example, messages that relate to a particular application might be included in a single category.

**cmf** See compiled message flow.

**collective**

A set of brokers that are fully interconnected and form part of a multi-broker network for publish/subscribe applications.



**compiled message flow (cmf)**

A message flow that has been compiled to prepare it for deployment to the broker. A cmf file is sent to the broker within a bar file.

**complex element**

A named structure that contains simple elements within the message. Complex elements can contain other complex elements, and can also contain groups. The content of a complex element is defined by a complex type. See also simple element.

**complex type**

A structure within a message. A complex type contains elements, attributes, and groups organized into a hierarchy. See also simple type.

**component**

A set of runtime processes that perform a specific set of functions. A component is a broker, a Configuration Manager, a Database Instance Manager, or a User Name Server.

**component directory**

In z/OS, the root directory of the component's runtime environment.

**component name**

The external name of a component. Each component requires a name, which is used, for example, in the workbench and in commands.

**component PDSE**

In a z/OS environment, a PDSE that contains jobs to define resources to DB2, WebSphere MQ, and the WebSphere Message Broker started task. See partitioned data set.

**configuration**

In a broker domain, the brokers, execution groups, deployed message sets, deployed message flows, and defined topics and access control lists.

**Configuration Manager**

The component that provides an interface between the workbench and a set of runtime brokers. It provides brokers with their initial configuration, and updates them with any subsequent changes. It maintains the broker domain configuration.

**Configuration Manager Proxy**

An application programming interface that your applications can use to control broker domains through a remote interface to the Configuration Manager.

**connection**

See message flow node connection. For broker-to-broker connections, see publish/subscribe topology.

**content-based filter**

In publish/subscribe, an expression that is included as part of a subscription to determine whether a publication message is received based on its content. The expression can include wild cards.

**Custom Wire Format**

The physical representation of a message in the MRM domain that is composed of a number of fixed format data structures or elements, which are not separated by delimiter characters.

**CWF** See Custom Wire Format.

## D

### **Database Instance Manager**

On Windows, a network server that supports the creation, maintenance, and deletion of databases used by brokers in all installations on a single computer. Database support is limited to Derby and DB2. The Database Instance Manager is associated with a Windows service.

### **data element separation**

A delimiter sequence that defines how a TDS message is to be parsed. The following separation types are supported: data pattern separation, delimited separation, fixed length separation, and tagged separation

### **DataFlowEngine (DFE)**

See execution group.

### **datagram**

A form of asynchronous messaging in which an application sends a message, but does not want a response. Also known as send-and-forget. Contrast with request/reply.

### **debugger**

See flow debugger.

### **deploy**

The process of transferring data to an execution group on a broker so that it can take effect in the broker domain. For deploying message flows and associated resources, the data is packaged in a broker archive (bar) file before being sent to the Configuration Manager, from where it is unpackaged and distributed appropriately.

**Derby** The database delivered by IBM Cloudscape Version 10.0, which is built on the Derby database from the Apache Software Foundation. Cloudscape does not modify Derby in any way, but provides additional function including installers. Derby database support is embedded in the broker component on Windows only.

### **destination list**

See local environment.

### **DataFlowEngine (DFE)**

See execution group.

### **distribution list**

A list of WebSphere MQ queues to which a message can be put with a single statement.

### **document type definition (DTD)**

The rules that specify the structure for a particular class of SGML or XML documents. The DTD defines the structure with elements, attributes, and notations, and it establishes constraints for how each element, attribute, and notation can be used within the particular class of documents. A DTD is analogous to a database schema in that the DTD completely describes the structure for a particular markup language.

**DTD** See document type definition.

## E

### **editor area**

The area in the workbench window where files are opened for editing.

**element**

A named piece of information, or a field, within a message, with a business meaning agreed by the applications that create and process the message. See also simple element and complex element.

**embedded message**

See multipart message.

**environment**

A structure within the message tree that is user-defined and can contain variable information that is associated with a message while it is being processed by a message flow.

**ESM** See external security manager.

**ESQL** See extended SQL.

**ESQL data type**

A characteristic of an item of data that determines how that data is processed. ESQL supports six data types (Boolean, datetime, null, numeric, reference, and string). Data that is retrieved from a database or is defined in a message model is mapped to one of these basic ESQL types when it is processed in ESQL expressions.

**ESQL field reference**

A sequence of values, separated by periods, that identify a specific field (which might be a structure) within a message tree or a database table. An example of a field reference is `Body.Invoice.InvoiceNo`.

**ESQL function**

A single ESQL expression that calculates a resultant value from a number of given input values. The function can take input parameters but has no output parameters; it returns to the caller the value that results from the implementation of the expression. The ESQL expression can be a compound expression such as `BEGIN END`.

**ESQL module**

A sequence of declarations that define MODULE-scope variables and their initialization, and a sequence of subroutine (function and procedure) declarations that define a specific behavior for a message flow node. A module must begin with the `CREATE node_type MODULE` statement and end with an `END MODULE` statement. The `node_type` must be one of `Compute`, `Database`, or `Filter`. The entry point of the ESQL code is the module scope procedure named `MAIN`.

**ESQL procedure**

A subroutine that has no return value. It can accept input parameters from and return output parameters to the caller.

**ESQL variable**

A local temporary field that is used to assist in the processing of a message.

**exception list**

A list of exceptions, with supporting information, that has been generated during the processing of a message.

**execution group**

A named process or set of processes within a broker in which message flows are run. The broker is guaranteed to enforce some degree of isolation between message flows in distinct execution groups because it ensures that they run in separate address spaces, or as unique processes.

An execution group process is also known as a DataFlowEngine (DFE); this term is typically used in problem determination scenarios (trace contents, diagnostic messages, and so on). A DFE is created as an operating system process, and has a one-to-one relationship with the named execution group. If more than one message flow runs within an execution group, multiple threads are created within the DFE process.

**extended SQL (ESQL)**

A specialized set of SQL functions and statements that are based on regular SQL, and extended with functions and statements that are unique to WebSphere Message Broker.

**extensible markup language (XML)**

A standard metalanguage for defining markup languages that was derived from, and is a subset of, Standard Generalized Markup Language (SGML).

**extensible stylesheet language (XSL)**

A language for specifying style sheets for XML documents. XSL Transformation (XSLT) is used with XSL to describe how an XML document is transformed into another document.

**external security manager (ESM)**

In a z/OS environment, a security product that performs security checking on users and resources. RACF is an example of an ESM.

**F**

**field reference**

See ESQL field reference

- filter** (1) An ESQL expression that is applied to the content of a message in a Filter node to determine how the message is processed.
- (2) An ESQL expression that is applied to the content of a publication message to determine whether the message matches certain criteria.

**flow debugger**

A facility to debug message flows that is provided in the Debug perspective in the workbench.

**format**

The definition of the internal structure of a message, in terms of the fields and the order of those fields. A format can be self-defining, in which case the message is interpreted dynamically when it is read.

**G**

**graphical user interface (GUI)**

A type of computer interface that presents a visual metaphor of a real-world scene, often of a desktop, by combining high-resolution graphics, pointing devices, menu bars and other menus, overlapping windows, icons, and the object-action relationship.

**group** A list of elements with information about how those elements can appear in a message. Groups can be ordered, unordered, or selective.

**GUI** See graphical user interface.

**I**

**IBM Runtime Environment for Java**

A subset of the IBM Developer Kit for the Java Platform that contains the

core executable files and other files that constitute the standard Java platform. The IBM Runtime Environment includes the Java virtual machine (JVM), core classes, and supporting files.

**IBM Software Developer Kit for Java**

A software package that can be used to write, compile, debug, and run Java applets and applications.

**IDOC domain**

The message domain that includes all messages that are exchanged between the broker and SAP R3 clients across the MQSeries link for R/3. Messages in this domain are processed by the IDOC parser. See also BLOB domain, JMS domain, MIME domain, MRM domain, XML domain, and XMLNS domain.

**IDOC parser**

A program that interprets a bit stream or tree that represents a message that belongs to the IDOC domain, and generates the corresponding tree from the bit stream on input, or the bit stream from the tree on output.

**implementation function**

A function written for a user-defined node or message parser; also known as a callback function.

**input node**

A message flow node that represents a source of messages for a message flow or subflow. See also output node.

**installation directory**

In a z/OS environment, a file system into which all product data is installed, and from which it is referenced and retrieved during the customization phase.

**J**

**Java Database Connectivity (JDBC)**

An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access. See also Open Database Connectivity.

**Java Message Service (JMS)**

An application programming interface that provides Java language functions for handling messages. See also Application Messaging Interface (AMI) and Message Queue Interface (MQI). Applications using JMS connect to the broker using either WebSphere MQ Real-time Transport or WebSphere MQ Multicast Transport.

**JCL** See Job Control Language

**JDBC** See Java Database Connectivity.

**JMS** See Java Message Service.

**JMS domain**

The message domain that includes all messages that are produced by the WebSphere MQ implementation of the Java Message Service standard. These messages, which have a message type of either JMSMap or JMSStream, are supported in the same way as messages in the XML domain and are parsed by the XML parser. See also BLOB domain, IDOC domain, MIME domain, MRM domain, XML domain, and XMLNS domain.

## **Job Control Language**

Job Control Language (JCL) comprises a set of Job Control Statements that are used to define work requests called jobs. JCL tells the operating system what program to run, and defines its inputs and outputs.

## **L**

**LIL** See loadable implementation library.

## **loadable implementation library (LIL)**

The implementation module for a node or parser written in C. This library file is implemented in the same way as a dynamic link library, but has a file extension of .lil rather than .dll.

## **local environment**

A structure within the message tree that contains broker and, optionally, user information associated with a message while it is being processed by a message flow.

In previous releases, the local environment structure was known as the Destination list; the latter term is retained for compatibility.

## **local error log**

A generic term that refers to the logs to which WebSphere Message Broker writes records on the local system. Also known as the system log.

## **M**

**map** (1) A complete transformation that has source objects that define the structure of the inputs and target objects that define the structure of the outputs. A map is represented as a .msgmap file.

(2) To associate a source to a target in a message map.

## **mapping**

A target value expression.

## **mapping cardinality**

The granularity of the way in which message elements are mapped from message source to message target. For example:

- One-to-one: associates a single source with a single target
- One-sided: associates a value with a target
- Many-to-one: associates multiple sources with a single target

## **message**

A communication that is sent from a person or program to another person or program. In WebSphere Message Broker, a message can be modeled by a message definition which describes the structure and content of the message. Messages must have a structure and format which is agreed by the sending and receiving applications.

## **message broker**

See broker.

## **Message Brokers Toolkit**

The WebSphere Message Broker development environment that integrates with IBM Rational Application Developer which is based on the IBM WebSphere Eclipse Platform. Also known as the workbench.

## **message definition**

A logical description of a message. A message definition is a structured collection of simple elements.

**message definition file**

A file that contains the messages, elements, types, and groups that make up a message set.

**message dictionary**

A data structure that describes all of the messages in a message set in a form suitable for deployment to a broker.

**message domain**

A grouping of messages that share certain characteristics. A message domain has an associated parser that interprets messages that are received and generated by a broker. WebSphere Message Broker supports messages in the BLOB domain, IDOC domain, JMS domain, MIME domain, MRM domain, and XML domain. You can create additional parsers known as user-defined parsers to support messages that do not conform to the supported domains.

**message element aggregation**

A mapping in which all the repeatable elements in one instance are mapped to another instance. It is not possible to map the repeatable elements themselves, only the instances. This aggregation is useful when mapping all possible inputs to one or more outputs, and can be used for copying an array, or for assigning a scalar, such as a summation. Use message element aggregation when the following conditions are met:

- A single source and target are selected.
- Source and target are of simple numeric type.
- The source repeats.

**message flow**

A sequence of processing steps that run in the broker when an input message is received. A message flow is created in the workbench by including a number of message flow nodes that each represents a set of actions that define a processing step. The connections in the flow determine which processing steps are carried out, in which order, and under which conditions. A message flow must include an input node that provides the source of the messages that are processed. Message flows are then ready to deploy to a broker for execution. See also subflow.

**message flow node**

A processing step in a message flow, also called a message processing node. A message flow node can be a built-in node, a user-defined node, or a subflow node.

**message flow node connection**

An entity that connects an output terminal of one message flow node to an input terminal of another. A message flow node connection represents the flow of control and data between two message flow nodes.

**message model**

A definition of a message format that is used by applications. You define a message model in the workbench.

**message parser**

A program that interprets the bit stream of an incoming message and creates an internal representation of the message in a tree structure, and that regenerates a bit stream for an outgoing message from the internal representation.

**message processing node**

See message flow node.



**Message Queue Interface (MQI)**

The programming interface that is provided by WebSphere MQ queue managers. The programming interface allows application programs to access message queuing services. See also Application Messaging Interface (AMI) and Java Message Service (JMS). Applications that use the MQI, connect to the broker using WebSphere MQ Enterprise Transport.

**message set**

A container; a logical grouping of messages and associated message resources (elements, types, groups).

**message set documentation**

A human-readable form of message definitions that you have created in the workbench.

**message set project**

A specialized container in which you create and maintain all the resources that are associated with a message set.

**message template**

A named and managed entity that represents the format of a particular message. Message templates represent a business asset of an organization.

**message tree**

The logical tree structure that represents the content and structure of a message in the broker. The message tree is created by a message parser from the input message received by a message flow.

**message type**

The logical structure of the data within a message; for example, the number and location of character strings.

**metadata**

The data that describes the characteristic of stored data.

**MIME** See Multipurpose Internet Mail Extensions.

**MIME domain**

The message domain that includes all messages that conform to the MIME standard. See also BLOB domain, IDOC domain, JMS domain, MRM domain, XML domain, and XMLNS domain.

**MIME parser**

A program that interprets a bit stream or tree that represents a message that belongs to the MIME domain, and generates the corresponding tree from the bit stream on input, or the bit stream from the tree on output.

**MQI** See Message Queue Interface.

**MQIsdp**

See SCADA device protocol.

**MQRFH**

An architected message header that is used to provide metadata for the processing of a message. This header is supported by the WebSphere MQ (MQSeries) Publish/Subscribe SupportPac.

**MQRFH2**

An extended version of MQRFH, providing enhanced function in message processing.

**MRM** The name given to the domain and parser that are associated with



messages that are modeled in the workbench. MRM stands for Message Repository Manager and is used only to identify the MRM parser and MRM domain.

**MRM domain**

The message domain that includes all messages that are modeled in the workbench. Message models can be created to represent a wide range of message types, with one or more optional physical formats. Messages in the MRM domain are processed by the MRM parser. See also BLOB domain, IDOC domain, JMS domain, MIME domain, XML domain, and XMLNS domain.

**MRM parser**

A program that interprets a bit stream or tree that represents a message that belongs to the MRM domain, and generates the corresponding tree from the bit stream on input, or the bit stream from the tree on output. Its interpretation depends on the physical format that you have associated with the input or output message.

**multilevel wild card**

A wild card that can be specified in subscriptions to match any number of levels in a topic.

**multipart message**

A message that contains one or more other messages within its structure. The contained message is sometimes referred to as an embedded message.

**Multipurpose Internet Mail Extensions**

An Internet standard that allows different forms of data, including video, audio, or binary data, to be attached to e-mail without requiring translation into ASCII text.

**N**

**namespace**

In XML, a uniform resource identifier (URI) that provides a unique name to associate with all the elements and type definitions in a schema. XML instance documents and XML schemas can make use of namespaces.

**node** (1) An endpoint or junction used in a message flow. See message flow node.

(2) An element in a message mapping tree. See tree node.

**O**

**ODBC**

See Open Database Connectivity.

**Open Database Connectivity (ODBC)**

A standard application programming interface (API) for accessing data in both relational and non-relational database management systems. Database applications that use this API can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and a different programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group.

**output node**

A message flow node that represents a point at which messages leave the message flow or subflow. See also input node.

## P

**parser** See message parser.

**partitioned data set (PDS, PDSE)**

In a z/OS environment, a data set in direct-access storage that is divided into partitions, which are called members. A partitioned data set (extended) (PDSE) is an extension to a PDS that contains an indexed directory in addition to the members.

**PDS, PDSE**

See partitioned data set.

**perspective**

A group of views that show various aspects of the resources in the workbench. The user can switch perspectives, depending on the task at hand, and customize the layout of views and editors within the perspective. See also view.

**physical format**

The physical representation of a message within the bit stream. The supported physical formats are Custom Wire Format, XML Wire Format, and Tagged/Delimited String Format.

**point-to-point**

A style of messaging application in which the sending application knows the destination of the message. Contrast with publish/subscribe.

**predefined element and message**

An element or message for which a matching definition exists in the message model. See also self-defining element and message.

**principal**

An individual user ID (for example, a login ID) or a group. A group can contain individual user IDs and other groups, to the level of nesting that is supported by the underlying facility.

**property**

A characteristic that, as one of a set of characteristics, defines the values and behaviors of objects in the workbench. For example, message flow nodes and deployed message flows have properties.

**publication**

A piece of information about a specified topic that is available to a broker in a publish/subscribe system.

**publication node**

An end point of a specific path through a message flow to which a client application subscribes, identified to the client by its subscription point.

**publisher**

An application that makes information about a specified topic available to a broker in a publish/subscribe system.

**publish/subscribe**

A style of messaging application in which the providers of information (publishers) are de-coupled from the consumers of that information (subscribers) using a broker. See also topic. Contrast with point-to-point messaging.

**publish/subscribe topology**

The brokers, the collectives, and the connections between them, that support publish/subscribe applications in the broker domain.

## Q

**queue** A WebSphere MQ object to which message queuing applications can put messages, and from which message queuing applications can get messages.

**queue manager**

A system program that provides queuing services to applications. A queue manager provides an application programming interface (the MQI) that enables programs to access messages on the queues that the queue manager owns.

## R

**request/reply**

A type of messaging application in which a request message is used to request a reply from another application. Contrast with datagram.

**resource**

A file of any type that exists in the workbench. You can view and edit a resource in the Broker Development view (previously called the Resource Navigator view) in the workbench.

**Resource Recovery Services (RRS)**

A z/OS facility that provides two-phase sync point support across participating resource managers.

**retained publication**

A published message that is kept at the broker for propagation to clients that subscribe in the future.

**RRS** See Resource Recovery Services.

## S

**SCADA**

See Supervisory, Control, And Data Acquisition.

**SCADA device protocol (MQIsdp)**

A protocol that implements the WebSphere MQ Telemetry Transport to connect SCADA devices to the broker.

**schema**

See XML Schema.

**self-defining element and message**

An element or message for which no matching definition exists in the message model. For example, a message that is coded in XML is self-defining. See also predefined element and message.

**send-and-forget**

See datagram.

**simple element**

A field in a message that is based on a simple type. A simple element can repeat, and it can define a default or a fixed value. See also complex element.

**simple type**

A characteristic of a simple element that defines the type of data within a message (for example, string, integer, or float). A simple type can have value constraints which place limits on the values of any simple elements based on that simple type. See also complex type.

**single-level wild card**

A wild card that can be specified in subscriptions to match a single level in a topic.

**SQL** See Structured Query Language.

**stream**

A method of topic partitioning that is used by applications that connect to WebSphere MQ Publish/Subscribe brokers.

**Structured Query Language (SQL)**

A standardized programming language that is used to define and manipulate data in a relational database. ESQL, the language that is used by WebSphere Message Broker, is based on SQL, and has many similar constructs.

**style sheet**

A specification of formatting instructions that, when applied to structured information, provides a particular rendering of that information (for example, online or printed). Different style sheets can be applied to the same piece of structured information to produce different presentations of the information.

**subflow**

A sequence of processing steps, implemented by message flow nodes, that is designed to be embedded in a message flow or in another subflow. A subflow must include at least one Input or Output node. A subflow can be started by a broker only as part of the message flow in which it is embedded, and therefore cannot be deployed.

**subflow node**

A message flow node that represents a subflow.

**subscriber**

An application that requests information about a specified topic from a publish/subscribe broker.

**subscription**

A record that contains the information that a subscriber passes to its local broker to describe the publications that it wants to receive.

**subscription filter**

A predicate that specifies the subset of messages that are to be delivered to a particular subscriber.

**subscription point**

The name that a subscriber uses to request publications from a particular set of publication nodes. It is the property of a publication node that differentiates that publication node from other publication nodes in the same message flow.

**substitution group**

An XML Schema feature that provides a means of substituting one element for another in an XML message. A substitution group contains a list of global elements that can appear in place of another global element, called the head element.

**Supervisory, Control, And Data Acquisition (SCADA)**

A term used to describe any form of remote telemetry system that is used to gather data from remote sensor devices (for example, flow rate meters on an oil pipeline) and for the near real time control of remote equipment

(for example, pipeline valves). These devices communicate with the broker using the SCADA device protocol (MQIsdp).

**system log**

See local error log.

**T**

**Tagged/Delimited String (TDS) Format**

The physical representation of a message in the MRM domain that has a number of data elements separated by tags and delimiters.

**TDS Format**

See Tagged/Delimited String Format.

**terminal**

The point at which one node in a message flow is connected to another node. You can connect terminals to control the route that a message takes, dependent on the outcome of the operation that is performed on that message by the node.

**topic** A character string that describes the nature of the data that is published in a publish/subscribe system.

**topic based subscription**

A subscription specified by a subscribing application that includes a topic that filters publications.

**topic security**

The application of ACLs to one or more topics to control subscriber access to published messages.

**topology**

See publish/subscribe topology.

**transform**

A defined way in which a message of one format is converted into one or more messages of another format.

**tree node**

An element in a mapping tree; a container for the mapping type such as an MRM message, database table, a column, or a basic element.

**type**

A characteristic of an element that describes its data content. See also simple type and complex type.

**U**

**Unicode Transformation Format, 8-bit encoding form (UTF-8)**

A transformation format that is designed for ease of use with existing ASCII-based systems. UTF-8 is an encoding of Unicode character strings that optimizes the encoding of ASCII characters in support of text-based communication.

**uniform resource identifier (URI)**

An encoded address that represents any resource, such as an HTML document, image, video clip, or program, on the Web; a URI is an abstract superclass compared with a Uniform resource locator or a Uniform resource name, which are concrete entities.

**uniform resource locator (URL)**

A sequence of characters that represent information resources on a computer or in a network such as the Internet. This sequence of characters includes:

- The abbreviated name of the protocol that is used to access the information resource
- The information that is used by the protocol to locate the information resource

A Web server typically maps the request portion of the URL to a path and file name. Also known as universal resource locator.

**uniform resource name (URN)**

A name that uniquely identifies a Web service to a client.

**URI** See uniform resource identifier.

**URL** See uniform resource locator.

**URN** See uniform resource name.

**user-defined extension**

An optional component that is designed by the user to extend the functionality of WebSphere Message Broker. A user-defined extension can be either a node or a message parser. See also user-defined node and user-defined parser.

**user-defined node**

An extension to the broker that provides a new message flow node in addition to those that are supplied with the product. See also implementation function and utility function.

**user-defined parser**

An extension to the broker that provides a new message parser in addition to those that are supplied with the product. See also implementation function and utility function.

**User Name Server**

A component that interfaces with operating system facilities to determine valid users and groups.

**UTF-8** See Unicode Transformation Format.

**utility function**

A function provided by the broker that can be used by developers who write user-defined nodes or parsers.

**V****value constraint**

A limit that sets a restriction on the values that a simple type can represent.

**view** A display in the workbench in which you can navigate and edit your information and resources. For example, the Broker Development view (previously called the Resource Navigator view) enables you to view and edit your project files. See also perspective.

**W****warehouse**

A persistent, data store for historical events (or messages). The Warehouse

node within a message flow supports the recording of information in a database for subsequent retrieval and processing by other applications.

**Web service**

A modular application that performs specific tasks and is accessible through open protocols like HTTP and SOAP.

**Web Services Description Language (WSDL)**

An XML-based specification for describing networked services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. A WSDL document enables a Web services client to invoke a Web service using the messages defined in a message definition.

**WebSphere MQ Enterprise Transport**

A transport protocol supported by WebSphere Message Broker that enables WebSphere MQ application clients to connect to brokers.

**WebSphere MQ Everyplace**

A generally available WebSphere MQ product that provides proven WebSphere MQ reliability and security for mobile and wireless devices. WebSphere MQ Everyplace applications connect to the broker using WebSphere MQ Mobile Transport.

**WebSphere MQ Mobile Transport**

A transport protocol supported by WebSphere Message Broker that enables WebSphere MQ Everyplace application clients to connect to brokers.

**WebSphere MQ Multicast Transport**

A transport protocol supported by WebSphere Message Broker that enables dedicated JMS application clients to connect to brokers. This protocol is optimized for high volume, one-to-many publish/subscribe topologies.

**WebSphere MQ Real-time Transport**

A transport protocol supported by WebSphere Message Broker that enables dedicated JMS application clients to connect to brokers.

**WebSphere MQ Telemetry Transport**

A transport protocol supported by WebSphere Message Broker that enables SCADA devices to connect to brokers. This protocol is a lightweight publish/subscribe protocol that flows over TCP/IP that uses a subset of UTF-8.

**WebSphere MQ Web Services Transport**

A transport protocol supported by WebSphere Message Broker that enables HTTP compliant application clients to connect to brokers.

**wild card**

A character that can be specified in subscriptions to match a range of topics. See also multilevel wild card and single-level wild card.

**workbench**

See Message Brokers Toolkit

**working set**

A logical collection of application projects, that you can use to limit the number of resources that are displayed in the Broker Development view. See also active working set.

**World Wide Web Consortium (W3C)**

An international industry consortium set up to develop common protocols to promote the evolution and interoperability of the World Wide Web.

**WSDL**

See Web Services Description Language.

**W3C** See World Wide Web Consortium.

**X**

**XML** See extensible markup language.

**XML domain**

The message domain that includes all messages that conform to the W3C XML standard. Messages in this domain are processed by the XML parser. See also BLOB domain, IDOC domain, JMS domain, MIME domain, MRM domain, and XMLNS domain.

**XML parser**

A program that interprets a bit stream or tree that represents a message that belongs to the XML domain and JMS domains, and generates the corresponding tree from the bit stream on input, or the bit stream from the tree on output. The bit stream is a representation of an XML file.

**XMLNS domain**

An extension of the XML domain that contains messages that conform to the same standard and that can exploit the namespaces feature of the XML specification. Messages in this domain are processed by the XMLNS parser. See also BLOB domain, IDOC domain, JMS domain, MIME domain, MRM domain, and XML domain.

**XMLNS parser**

A program that interprets a bit stream or tree that represents a message that belongs to the XMLNS domain, and generates the corresponding tree from the bit stream on input, or the bit stream from the tree on output. The bit stream is a representation of an XML file.

**XML Path Language (XPath)**

An XSL sublanguage designed to uniquely identify or address parts of a source XML document, for use with XSLT. XPath also provides basic facilities for the manipulation of strings, numbers, and Boolean values.

**XML schema**

An international standard that defines a language for describing the structure of XML documents. An XML schema formally describes and constrains the content of XML documents by indicating which elements are allowed and in which combinations. (An XML Schema is an alternative to a document type definition (DTD), and can be used to extend functionality in the areas of data typing, inheritance, and presentation.) The XML Schema language is ideally suited to describing the messages that flow between business applications, and is widely used in the business community for this purpose.

**XML Schema Definition Language (XSD)**

A language for describing XML files that contain XML schema.

**XML Wire Format**

The physical representation of a message in the MRM domain that can be parsed as XML.

**XPath** See XML Path Language.

**XSD** See XML Schema Definition Language (XSD).

**XSL** See extensible stylesheet language.



---

## Part 4. Appendixes



---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032,  
Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire,  
England  
SO21 2JN*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information includes examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) *(your company name)* (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *\_enter the year or years\_*. All rights reserved.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX	CICS	Cloudscape
DB2	DB2 Connect	DB2 Universal Database
developerWorks	Domino	
Everyplace	FFST	First Failure Support Technology
IBM	IBMLink	IMS
IMS/ESA	iSeries	Language Environment
Lotus	MQSeries	MVS
NetView	OS/400	OS/390
POWER	pSeries	RACF
Rational	Redbooks	RETAIN
RS/6000	SupportPac	S/390
Tivoli	VisualAge	WebSphere
xSeries	z/OS	zSeries

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



---

# Index

## A

- access control lists
  - User Name Server 74
- accessibility 76
  - information centre 4
  - keyboard 77
  - keyboard shortcuts 115
  - shortcut keys 77
- administration 67
  - Broker Administration perspective 81
- API, Configuration Manager 69
- archive, Broker Archive editor 91

## B

- bar files
  - Broker Archive editor 91
- Broker Administration perspective 81
- Broker Application Development perspective 83
- Broker Archive editor 91
- broker archive files
  - Broker Archive editor 91
- broker domains 67
  - diagram 67
- Broker Topology editor 93
- brokers 70
  - broker domain 67
  - configuration details 68
  - execution groups 72
  - managing 93
  - system management 71
- business integration, WebSphere MQ 37

## C

- changes in this release 39
  - Version 5.0 52
  - Version 5.0 fix packs 55
  - Version 6.0 39
  - Version 6.0 fix packs 49
  - Version 6.0.2.0 45
- CICS 37
- client environment 60
- code pages
  - editor preferences 91
- components
  - broker domains 67
  - brokers 70
  - Configuration Manager 68
  - execution groups 72
  - User Name Server 74
- Configuration Manager 68
  - API 69
- Configuration Manager Proxy 69
- context-sensitive help 60
- conventions, resource names 115

## D

- DataFlowEngine 72
- Debug perspective 87
- development
  - Broker Application Development perspective 83
  - Plug-in Development perspective 90
- development repository 66
- disability 76
- display, high-contrast 76

## E

- Eclipse 60
- editor preferences, localized settings 91
- editors 62
  - Broker Archive 91
  - Broker Topology 93
  - ESQL 93
  - Event Log 96
  - localized settings 91
  - Message Category 97
  - Message Definition 98
  - Message Flow 102
  - Message Mapping 107
  - Message Node 108
  - Message set 109
  - preferences 91
  - Subscriptions Query 110
  - Topics Hierarchy 111
- environments
  - client 60
  - runtime 67
- ESQL
  - editor 93
  - files
    - sharing 91
- Event Log editor 96
- execution groups 72

## F

- files 63
  - message flows 112
  - message sets 113
  - plug-in development 114
  - sharing 91
- folders 63

## H

- help
  - information centre 4
  - sources 5
- help, context-sensitive 60
- high contrast display 76

## I

- icons
  - Broker Application Development perspective 83
- IMS/ESA 37
- infopops 60
- information centre 4
- installation
  - verifying 8
- introduction 37
  - administering applications 27
  - administration 30
  - application development concepts 9
  - broker administration 24
  - configuring a broker domain 24
  - configuring applications 27
  - deploying applications 22
  - message flow applications 9
  - message flows
    - developing 12
    - programming 14
  - message models
    - designing 18
    - developing 19
  - monitoring 30
  - new users 6
  - overview 6
  - planning a broker domain 24
  - previous users 33
  - problem determination 31
  - publish/subscribe
    - developing 21
  - scenarios 74
  - technical 57
  - testing applications 22
  - WebSphere Message Brokers 37

## K

- keyboard 77
- keyboard shortcuts 115

## L

- linking objects by name 66
- Lotus Domino 37

## M

- mappings
  - Message Mapping editor 107
- Message Brokers Toolkit 60
  - using 7
- Message Category editor 97
- message category files
  - editing 97
- Message Definition editor 98
  - Outline view 99
  - Overview editor 100

- Message Definition editor (*continued*)
  - Properties editor 101
- message definition files
  - Message Definition editor 98
- Message Flow editor 102
- message flows
  - Broker Application Development perspective 83
  - Debug perspective 87
  - execution groups 72
  - Mapping editor 107
  - projects and files 112
- Message Mapping editor 107
- Message Node editor 108
- Message set editor 109
- message set files
  - creating 109
  - editing 109
- message sets
  - Broker Application Development perspective 83
  - projects and files 113
- Microsoft Exchange 37

## N

- name, linking by 66
- naming resources 115
- navigation
  - information centre 4
- new function and capabilities 39
  - Version 5.0 52
  - Version 5.0 fix packs 55
  - Version 6.0 39
  - Version 6.0 fix packs 49
  - Version 6.0.2.0 45

## O

- objects, linking by name 66
- overview 57
  - scenarios 74

## P

- perspectives 61
  - Broker Administration 81
  - Broker Application Development 83
  - Debug 87
  - overview 81
  - Plug-in Development 90
  - switching 61
- Plug-in Development perspective 90
- plug-in development, projects and files 114
- preferences
  - editors 91
- projects 63
  - message flows
    - resource files 112
  - message sets
    - resource files 113
  - plug-in development 114
  - reference property 66
- publish/subscribe
  - User Name Server 74

## Q

- queues 70

## R

- reference, project 66
- repository, development 66
- resources 63
  - brokers 70
  - Configuration Manager 68
  - message flows 112
  - message sets 113
  - naming rules 115
  - plug-in development 114
  - types 112
  - User Name Server 74
- rules
  - resource names 115
- runtime
  - components 67
  - environment 67

## S

- SAP/R3 37
- scenarios 74
  - WebSphere Message Brokers 74
- searching
  - information centre 4
- shortcut keys 77
- shortcuts, keyboard 115
- Subscriptions Query editor 110
- Supply Chain Management 37
- system management interfaces 71

## T

- text-only Quick Tour 57
- Toolkit 60
- Topics Hierarchy editor 111
- topology
  - broker 93
- trademarks 141

## U

- User Name Server 74
- user-defined nodes, editing 108
- UTF-8 91

## W

- WebSphere MQ 37
  - AMI 37
  - business integration 37
  - MQI 37
  - WebSphere MQ messaging 37
- workbench 60
  - Broker Administration perspective 81
  - Broker Application Development perspective 83
  - Debug perspective 87
  - Plug-in Development perspective 90
- working sets 63







Printed in USA