

WebSphere Event Broker



Introduction

Version 6 Release 0

WebSphere Event 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® Event 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. Product overview 1

Product overview. 3

WebSphere Message Brokers introduction. 3

WebSphere Event Broker Version 6.0 13

Accessibility features for WebSphere Event Broker 30

Part 2. Reference 31

Workbench 33

Perspectives in the Message Brokers Toolkit . . . 33

Editors in the Message Brokers Toolkit 39

Resource types in the Message Brokers Toolkit . . . 50

Message Brokers Toolkit keyboard shortcuts . . . 50

Glossary of terms and abbreviations 55

Part 3. Appendixes. 65

Appendix. Notices 67

Trademarks 69

Index 71

About this topic collection

This PDF has been created from the WebSphere Event Broker Version 6.0 (Fix Pack 4 update, March 2007) information center topics. Always refer to the WebSphere Event 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. Product overview

| | |
|---|----------|
| Product overview | 3 |
| WebSphere Message Brokers introduction. | 3 |
| Using WebSphere Message Brokers in your business | 4 |
| What's new in Version 6.0? | 5 |
| WebSphere Event Broker Version 6.0 | 13 |
| Create the run time. | 14 |
| Develop applications | 15 |
| Deploy applications to the run time | 15 |
| Publish/subscribe | 16 |
| Message Brokers Toolkit | 16 |
| Runtime environment | 22 |
| Accessibility features for WebSphere Event Broker | 30 |

Product overview

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

- “WebSphere Message Brokers introduction”
 - Quick Tour
 - “What’s new in Version 6.0?” on page 5
- “WebSphere Event Broker Version 6.0” on page 13
- “Accessibility features for WebSphere Event Broker” on page 30
- 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 Event Broker Version 6.0. See also "New function added in Version 6.0 fix packs" on page 9.

- Extended platform support
- Simplified installation and migration
- Improved Message Brokers Toolkit
- Improved performance and scalability
- Coexistence
- Driving message flows
-

- Improved management
- Improved security
- User-defined nodes
- 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 23.

The Message Brokers Toolkit can be installed on Linux (x86 platform) in addition to Windows; see “Message Brokers Toolkit” on page 16.

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 Event Broker Version 6.0 is easier than the installation of WebSphere Business Integration Event 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 Event Broker Version 2.1 and WebSphere Business Integration Event Broker Version 5.0 are supported. Both Version 2.1 and Version 5.0 can Coexist with WebSphere Event 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 Event 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 Message flows, which reduces the need for Eclipse perspective switching. For more information, see Flow debugger overview.

For more information about the Message Brokers Toolkit, see “Message Brokers Toolkit” on page 16.

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.
- Publish/subscribe functions have been improved.
- The storage requirements of the XML parsers have been reduced so that larger messages can be processed.

Coexistence

WebSphere Event 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 Event Broker Version 6.0 in a different location on the same computer, migrate your components and resources to WebSphere Event 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.

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.

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 Event 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

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

User-defined nodes

You can deploy user-defined nodes that have been created and supplied by WebSphere Message Broker Version 6.0 users, or by independent software vendors and other companies. Use these nodes in WebSphere Event Broker message flows to add to the function provided by the supplied or built-in nodes.

For more information, see *User-defined nodes*.

License management

WebSphere Event Broker supports a new form of license management using IBM Tivoli License Manager (ITLM), Version 2.1, across all the WebSphere Event 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 Event 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 5 topic introduces you to the main new function in WebSphere Event Broker Version 6.0, "New function added in Version 6.0 fix packs" on page 9 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.

- New Message Flow wizard
- Broker Development view (formerly the Resource Navigator view)
- Displaying projects in working sets
- Global toolbar in the Broker Application Development perspective
- New categories for nodes on the node palette
- Editing node properties
- Refreshing the contents of a broker archive

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.

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 19 and “Broker Application Development perspective” on page 35 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 19 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 35 for more information.

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 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.

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.

New function added in Version 6.0 fix packs

Whereas the “What’s new in Version 6.0?” on page 5 topic introduces you to the main new function in WebSphere Event 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 Event 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

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 Event 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 Event Broker plug-in now available for the IBM Support Assistant (ISA)

A WebSphere Event 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.

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 mqsichangeproperties 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 Event 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 Event 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)
- 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

New in Version 5.0

This topic introduces you to the main new function in WebSphere Business Integration Event Broker Version 5.0. See also New function added in Version 5.0 fix packs.

- New graphical development environment based on Eclipse
- Publish/subscribe enhancements
- Enhanced initial user experience
- Message flow accounting and statistics
- 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 Event Broker message flows (including publish/subscribe applications). See the “Message Brokers Toolkit” on page 16 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 16.
- 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 22.
- Improved editors make it easier for you to build your solution. See “Editors in the Message Brokers Toolkit” on page 39.
- You can configure some message flow properties at deployment time rather than at development time. See Configurable properties of a broker archive .

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.

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 Soccer Results sample sample.
- Explore the Samples Gallery.

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

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.

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

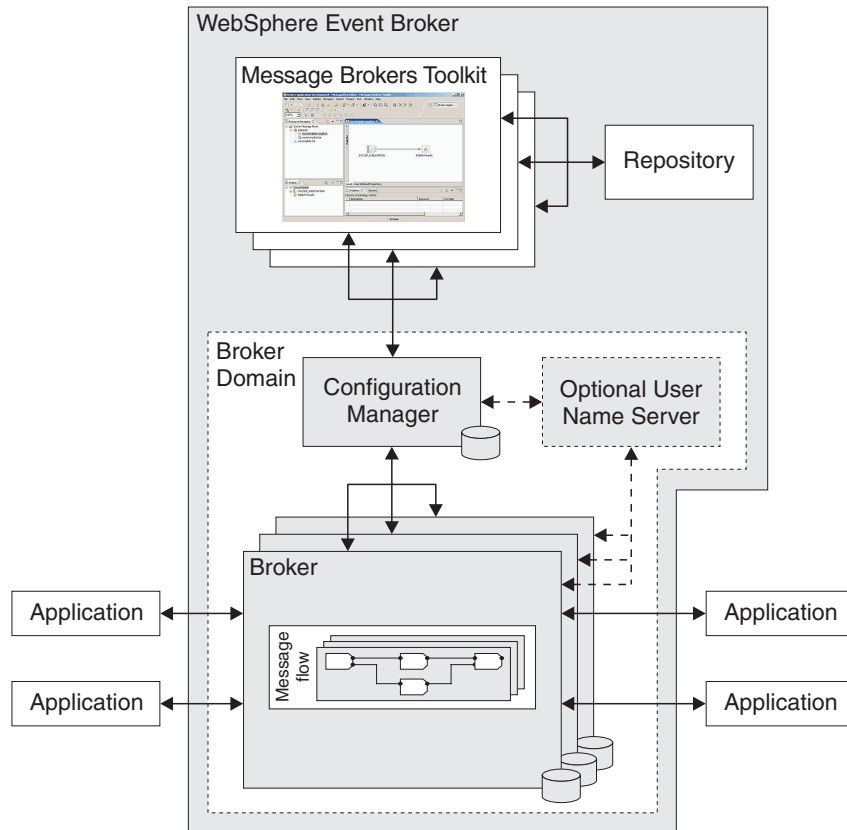
You can now control access to runtime resources by object as opposed to by group. See Authorization to access runtime resources.

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.

WebSphere Event Broker Version 6.0

WebSphere Event 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.



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 deploy user-defined nodes that have been created and supplied by WebSphere Message Broker Version 6.0 users or by independent software vendors and other companies.

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

Create the run time

The work of routing 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 using the workbench.

Different perspectives in the workbench are used to develop message flows 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 Broker JMS Transport

Deploy applications to the run time

When message flows 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, 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 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 between the workbench and brokers within its domain. WebSphere MQ messaging is used between the workbench, the Configuration Manager, and the brokers.

Publish/subscribe

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 flows. The Message Brokers Toolkit also communicates with one or more Configuration Managers, and you can use it to manage broker domains.

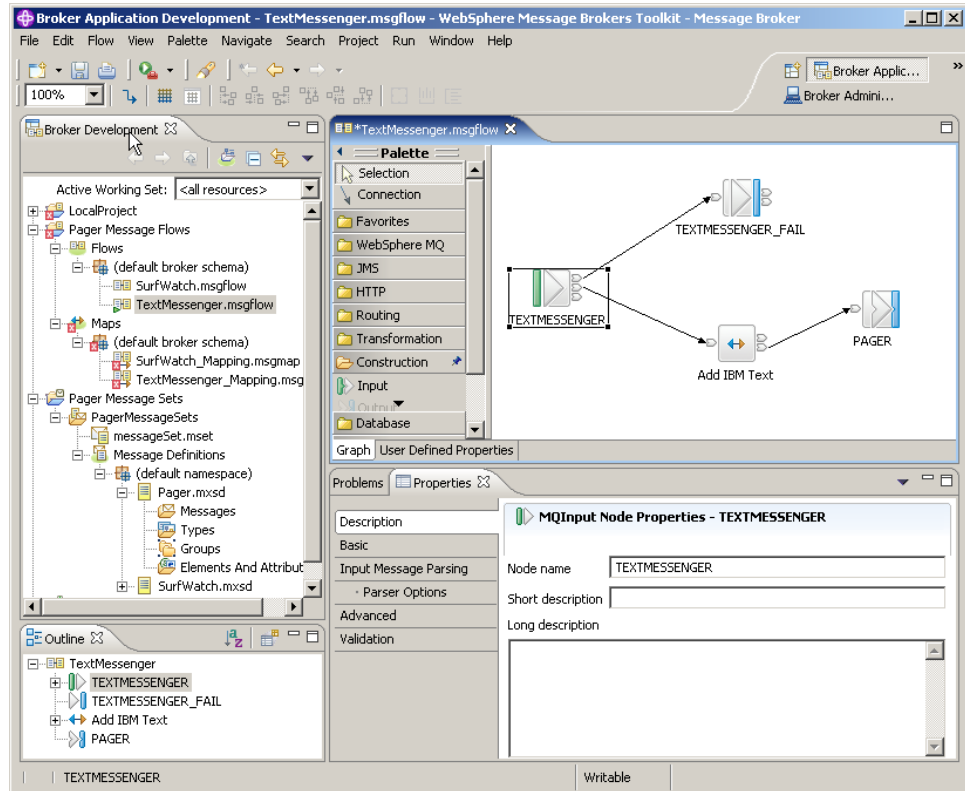
You can install the Message Brokers Toolkit on Windows and Linux (x86 platform).

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.

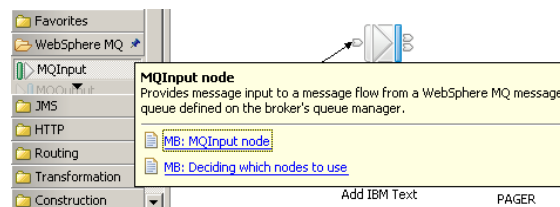
The following figure shows the Broker Application Development perspective with a message flow open in the Message Flow editor.



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.

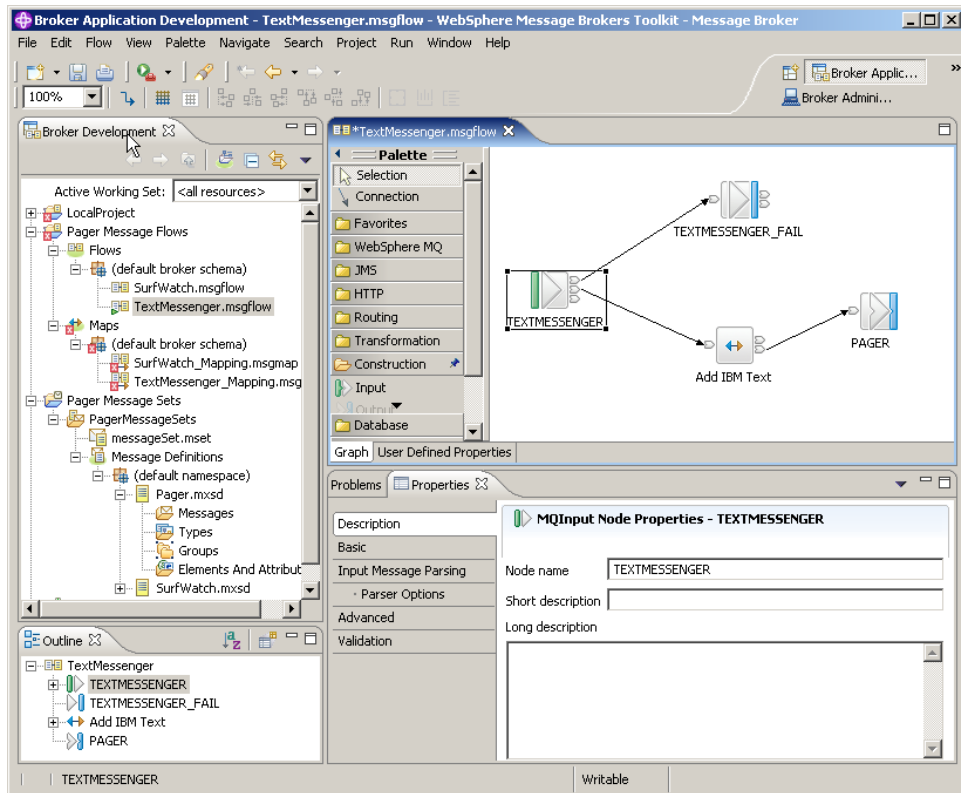


Perspectives

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 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.

Data perspective

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

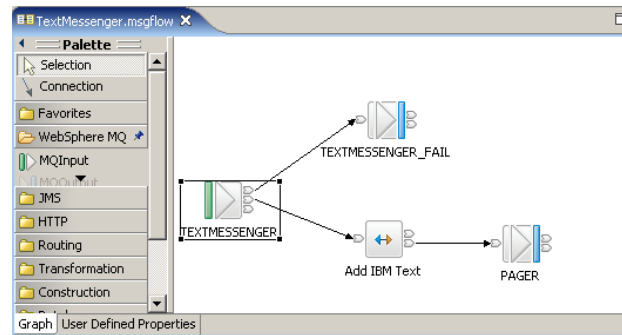
Editors

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.

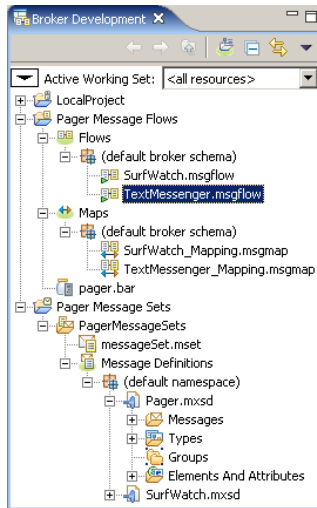
Resources

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, and broker archive files. File types are listed in “Resource types in the Message Brokers Toolkit” on page 50.

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.

Enqueues

This virtual folder is shown only when it contains .enqueue files, which are 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 50.

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 and add existing application projects to it.

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 Event 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 Event 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.

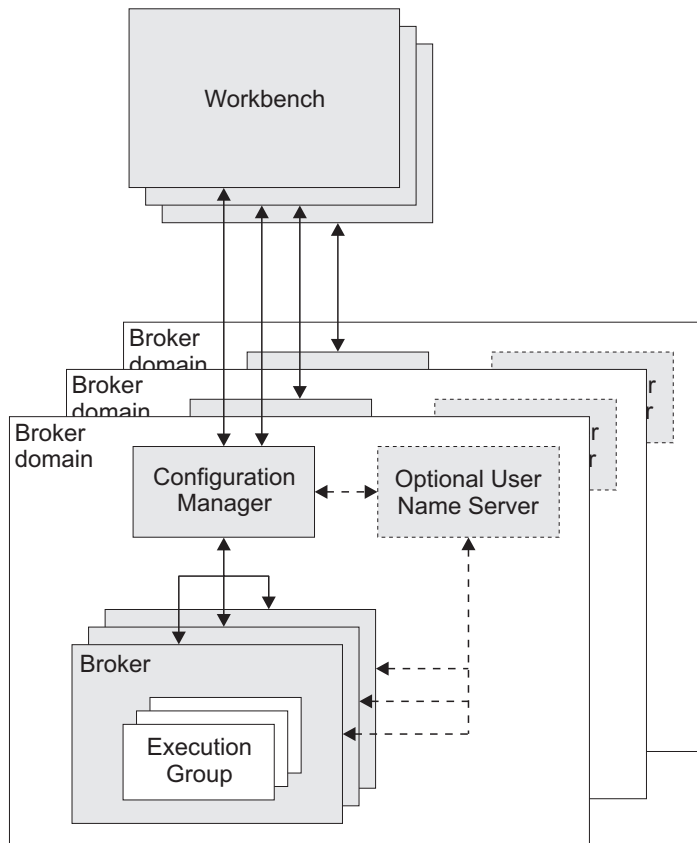
Because the Message Brokers Toolkit is based on the Eclipse platform, you can access Eclipse-supported repositories directly from the Message Brokers Toolkit.

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 26
- “Execution groups” on page 28
- “Broker domains”
- “Configuration Manager”
- “User Name Server” on page 29

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.

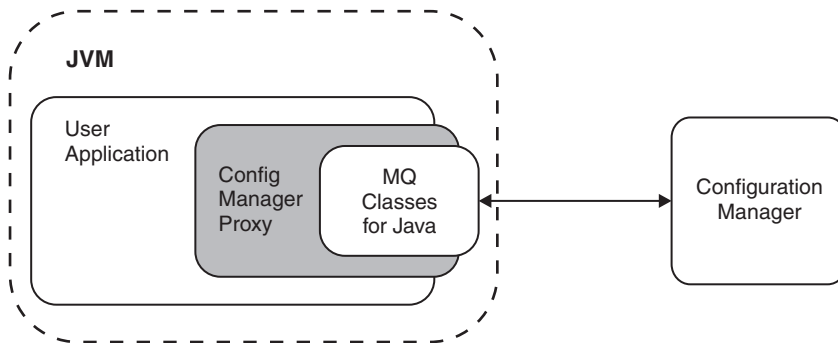
Configuration Manager Proxy:

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:

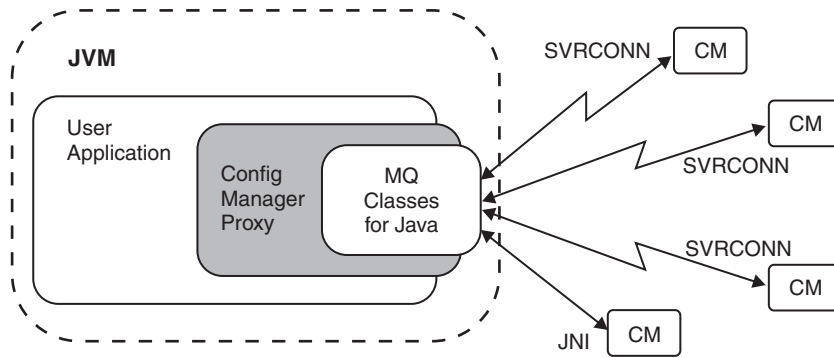
- Deploy BAR files, Publish/Subscribe topology, topic trees, and broker configuration.
- Modify the Publish/Subscribe topology; add and remove brokers, broker connections, and collectives.
- Create, modify, and delete execution groups
- Enquire and set status of objects in the domain (for example, run 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. 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 Event 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 Event Broker broker domain.

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

WebSphere Event Broker brokers publish event messages, using fixed topics, in response to configuration changes, state changes, and user actions such as subscription registrations. WebSphere Event 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 Event Broker broker domain.

You can develop or purchase system management adapters or customized administrative applications. These subscribe to the system management topics generated by WebSphere Event 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.

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.

Accessibility features for WebSphere Event 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 Event 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 Event 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 2. Reference

| | |
|---|-----------|
| Workbench | 33 |
| Perspectives in the Message Brokers Toolkit | 33 |
| Broker Administration perspective. | 33 |
| Broker Application Development perspective | 35 |
| Editors in the Message Brokers Toolkit | 39 |
| Broker Archive editor | 39 |
| Broker Topology editor | 40 |
| Event Log editor | 41 |
| Message Flow editor | 42 |
| Subscriptions Query editor | 47 |
| Topics Hierarchy editor | 49 |
| Resource types in the Message Brokers Toolkit | 50 |
| Message flow projects and files. | 50 |
| Rules for naming workspace objects | 50 |
| Message Brokers Toolkit keyboard shortcuts | 50 |
| | |
| Glossary of terms and abbreviations | 55 |

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 35

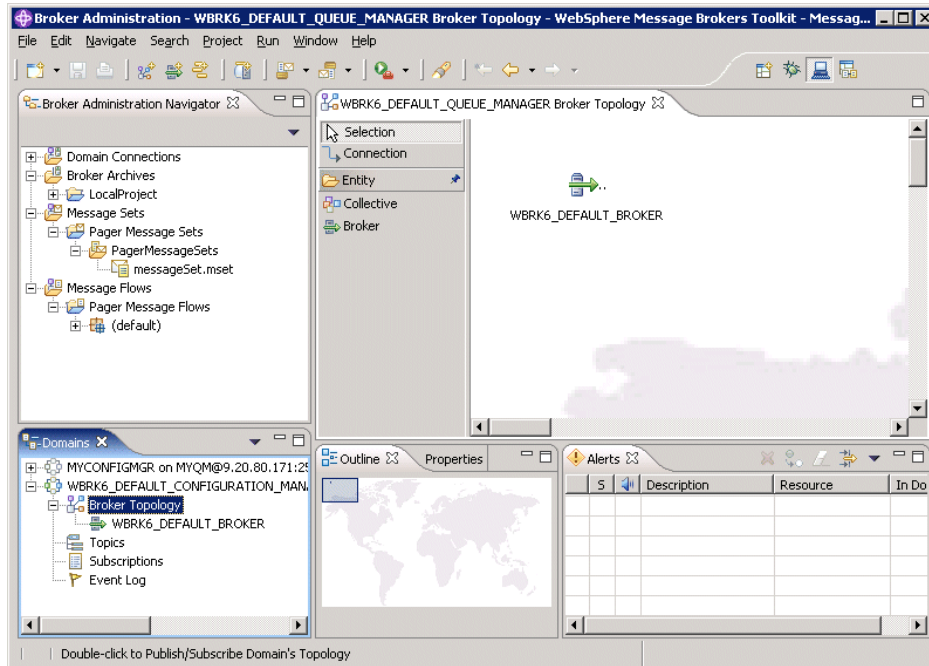
Broker Administration perspective

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 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 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.

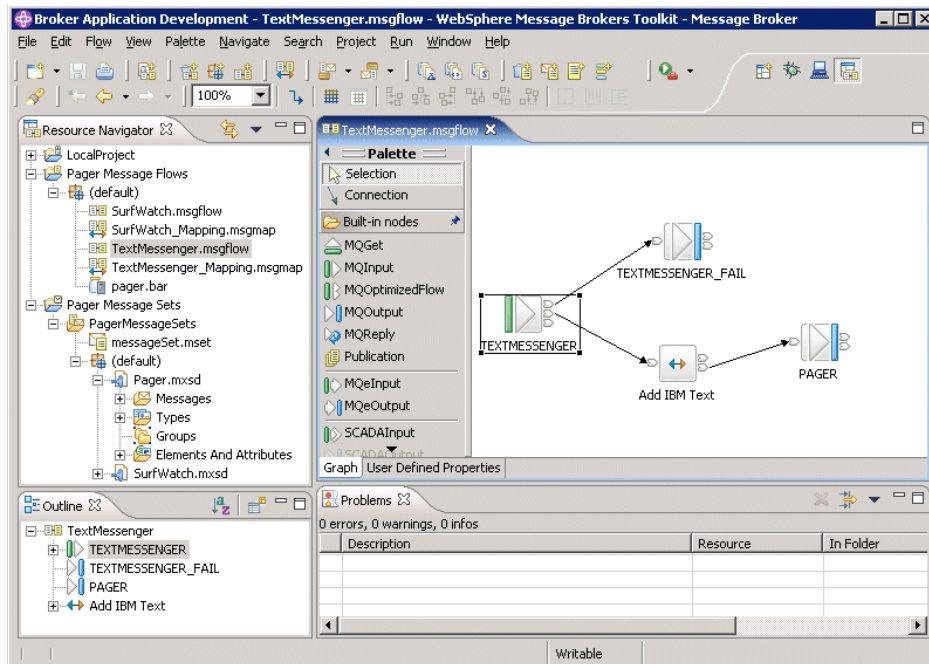
Broker Application Development perspective

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 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.

The Broker Development view displays a hierarchical view of the following message flow resource files:

- Message flow projects
- Message flows

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 19.

Other folders and files that are created for the Message Flow 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 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 19 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 Message Flow editor by default for working with the content of message flow files.

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 |
|  | 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 |

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”
- “Broker Topology editor” on page 40
- “Event Log editor” on page 41
- “Message Flow editor” on page 42
- “Subscriptions Query editor” on page 47
- “Topics Hierarchy editor” on page 49

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 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 |
|---|-------|---------------------------------------|
|  | Add | To add message flows to this archive. |

| Icon | Label | Action |
|---|----------------|---|
|  | Remove | To remove message flows from this archive. |
|  | Edit | To edit selected message flows in this archive. |
|  | Refresh | 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:

- .cmf files

You cannot edit the Version column.

You can use the `mqswireadbar` 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](#).

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 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 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. |

| Icon | Label | Action |
|---|--------------------------------|--|
|  | 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. |
|  | 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 Event 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 |

| Menu item | Keyboard shortcut |
|-------------------|-------------------|
| 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 50.

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 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

You can define more than one message flow, each in a separate .msgflow file, in the same message project.

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 Event 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 Event Broker terms and abbreviations that are used in this online information center.

The migration glossary lists differences in terminology between WebSphere Event 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.

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.

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.

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) 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 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 (message flows) that are defined within it.

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

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.

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 Event Broker started task. See partitioned data set.

configuration

In a broker domain, the brokers, execution groups, 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.

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.

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.

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.

DataFlowEngine (DFE)

See execution group.

distribution list

A list of WebSphere MQ queues to which a message can be put with a single statement.

E

editor area

The area in the workbench window where files are opened for editing.

ESM See external security manager.

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.

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).

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.

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.

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.

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.

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

local error log

A generic term that refers to the logs to which WebSphere Event Broker writes records on the local system. Also known as the system log.

M**message**

A communication that is sent from a person or program to another person or program. In WebSphere Event Broker, 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 Event 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 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 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.

metadata

The data that describes the characteristic of stored data.

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.

multilevel wild card

A wild card that can be specified in subscriptions to match any number of levels in a topic.

N

node An endpoint or junction used in a message flow. See message flow 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.

point-to-point

A style of messaging application in which the sending application knows the destination of the message. Contrast with publish/subscribe.

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.

send-and-forget

See datagram.

single-level wild card

A wild card that can be specified in subscriptions to match a single level in a topic.

stream

A method of topic partitioning that is used by applications that connect to WebSphere MQ Publish/Subscribe brokers.

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.

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**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.

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 node

An extension to the broker that provides a new message flow node in addition to those that are supplied with the product. A user-defined node cannot be developed in WebSphere Event Broker, but can be imported and deployed.

User Name Server

A component that interfaces with operating system facilities to determine valid users and groups.

UTF-8 See Unicode Transformation Format.

V

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

WebSphere MQ Enterprise Transport

A transport protocol supported by WebSphere Event 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 Event Broker that enables WebSphere MQ Everyplace application clients to connect to brokers.

WebSphere MQ Multicast Transport

A transport protocol supported by WebSphere Event 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 Event Broker that enables dedicated JMS application clients to connect to brokers.

WebSphere MQ Telemetry Transport

A transport protocol supported by WebSphere Event 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.

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

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.

W3C See World Wide Web Consortium.

X

XML See extensible markup language.

Part 3. 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 29
- accessibility 30
 - keyboard 30
 - keyboard shortcuts 51
 - shortcut keys 30
- administration 22
 - Broker Administration perspective 33
- API, Configuration Manager 25
- archive, Broker Archive editor 39

B

- bar files
 - Broker Archive editor 39
- Broker Administration perspective 33
- Broker Application Development perspective 35
- Broker Archive editor 39
- broker archive files
 - Broker Archive editor 39
- broker domains 23
 - diagram 22
- Broker Topology editor 40
- brokers 26
 - broker domain 23
 - configuration details 23
 - execution groups 28
 - managing 40
 - system management 27
- business integration, WebSphere MQ 3

C

- changes in this release 5
 - Version 5.0 12
 - Version 6.0 5
 - Version 6.0 fix packs 9
 - Version 6.0.2.0 8
- CICS 3
- client environment 16
- components
 - broker domains 23
 - brokers 26
 - Configuration Manager 23
 - execution groups 28
 - User Name Server 29
- Configuration Manager 23
 - API 25
- Configuration Manager Proxy 25
- context-sensitive help 16
- conventions, resource names 50

D

- DataFlowEngine 28

- development

- Broker Application Development perspective 35
- development repository 22
- disability 30
- display, high-contrast 30

E

- Eclipse 16
- editors 18
 - Broker Archive 39
 - Broker Topology 40
 - Event Log 41
 - Message Flow 42
 - Subscriptions Query 47
 - Topics Hierarchy 49
- environments
 - client 16
 - runtime 22
- Event Log editor 41
- execution groups 28

F

- files 19
 - message flows 50
- folders 19

H

- help, context-sensitive 16
- high contrast display 30

I

- icons
 - Broker Application Development perspective 35
- IMS/ESA 3
- infopops 16
- introduction 3
 - technical 13
 - WebSphere Message Brokers 3

K

- keyboard 30
- keyboard shortcuts 51

L

- linking objects by name 21
- Lotus Domino 3

M

- Message Brokers Toolkit 16
- Message Flow editor 42
- message flows
 - Broker Application Development perspective 35
 - execution groups 28
 - projects and files 50
- message sets
 - Broker Application Development perspective 35
- Microsoft Exchange 3

N

- name, linking by 21
- naming resources 50
- new function and capabilities 5
 - Version 5.0 12
 - Version 6.0 5
 - Version 6.0 fix packs 9
 - Version 6.0.2.0 8

O

- objects, linking by name 21
- overview 13

P

- perspectives 17
 - Broker Administration 33
 - Broker Application Development 35
 - overview 33
 - switching 17
- projects 19
 - message flows resource files 50
 - reference property 21
- publish/subscribe
 - User Name Server 29

Q

- queues 26

R

- reference, project 21
- repository, development 22
- resources 19
 - brokers 26
 - Configuration Manager 23
 - message flows 50
 - naming rules 50
 - types 50
 - User Name Server 29

- rules
 - resource names 50
- runtime
 - components 22
 - environment 22

S

- SAP/R3 3
- shortcut keys 30
- shortcuts, keyboard 51
- Subscriptions Query editor 47
- Supply Chain Management 3
- system management interfaces 27

T

- text-only Quick Tour 13
- Toolkit 16
- Topics Hierarchy editor 49
- topology
 - broker 40
- trademarks 69

U

- User Name Server 29

W

- WebSphere MQ 3
 - AMI 3
 - business integration 3
 - MQI 3
 - WebSphere MQ messaging 3
- workbench 16
 - Broker Administration perspective 33
 - Broker Application Development perspective 35
- working sets 19



Printed in USA