



IBM Software Group

WebSphere® Message Broker Overview

Introduction



@business on demand.

© 2006 IBM Corporation
Updated October 11, 2006

This presentation provides an introduction of WebSphere Message Broker.

Agenda

- WebSphere Message Broker Overview
- Components
- Summary and references



This presentation provides a brief overview of the message broker and its functions. This presentation will also cover the components of a message broker.

Section

WebSphere Message Broker Overview



The first section of the presentation provides an overview of the WebSphere Event Broker and WebSphere Message Broker and how you might use the broker in your enterprise.

WebSphere brokers

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.

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.



WebSphere Brokers help you integrate your business processes and applications by bringing together the data and information of your enterprise, allowing you to achieve business goals.

The brokers use WebSphere MQ to provides a secure, platform independent communications infrastructure. It provides you with the flexibility to apply intelligence to your business data as it moves through your enterprise.

WebSphere Message Broker capabilities

- Applications 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.
- Diverse applications can **exchange information in dissimilar forms**, with Message 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**.
- The 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 a broker gives your applications the flexibility to select which messages to receive based on topic or content-based filters.

Message Brokers can ensure that applications receive data in the proper format without knowing anything except their own conventions and requirements. The broker can do the transformation outside of the applications, allowing diverse applications to work in concert. This is valuable for legacy applications which do not need to be changed. It also means that changes in one area of the enterprise do not require reprogramming in other areas.

The Message Broker provides the framework supporting basic functions and user-defined enhancements which facilitate rapid development and modification of business processing rules that are applied to the messages in the system.

WebSphere Message Broker products

- **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** incorporates WebSphere Event Broker and extends its function to provide a powerful, point-to-point 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).
- **Rules and Formatter Extension** ensures continued use of existing business integration logic defined with Rules and Formatter Nodes from earlier versions of Message Broker.



There are three products available in the WebSphere domain. The Event Broker is a publish and subscribe product that provides a publish and subscribe function to clients connected using various transports. Using non-persistent messages provides a high-performance interaction.

The Message Broker includes the Event Broker and extends its function to provide a powerful message broker. The Message Broker is a point-to-point model. The messages are formed, routed and transformed to meet the business needs of the enterprise. A graphical user interface gives you development tools.

The Rules and Formatter Extension provides compatibility with earlier versions of Message Broker.

WebSphere Event Broker

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.



The publish/subscribe model allows distribution of real-time information from disparate sources through a centralized broker.

It allows you to publish a message to make it available to other applications. Other applications can choose to receive publications on specific topics, or that have specific content, or both.

The Event Broker provides a method to create topics and subscription points and provides access control functions.

WebSphere Message Broker

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 one 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.



The Message Broker allows you to manage the flow of information through your enterprise and integrate your applications.

In addition to the publish/subscribe functions of the Event Broker, the Message Broker allows you to:

Using rules based on the contents of the message, to route a message to multiple destinations

Transform the format of a message to accommodate the receiving application's requirements

Interact with a database by retrieving a message or storing a message

Modify the contents of a message

WebSphere Message Broker (cont.)

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

- 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.
- Systems can be extended to reach suppliers and customers by **meeting their interface requirements within the brokers**; can help improve the quality of interactions, allowing expeditious responses to changing or additional requirements.



Using the Message Broker to do transformations in a single place enables you to integrate your processes and applications. Since the transformation logic is in a single place, the Broker can help reduce the cost of maintenance and changes to applications.

The Message Broker also provides a mechanism for interfacing with applications outside of your enterprise; you would be able to transform and route data to your business partners and customers, meeting their format requirements.

WebSphere Message Broker platforms

- AIX®
- HP-UX
- Linux® (x86)
- Linux (zSeries)
- Solaris
- Windows®
- z/OS®

Always check the WebSphere Message Broker site for the latest updates:

<http://www-306.ibm.com/software/integration/wbimessagebroker/>



The WebSphere Event Broker and Message Broker products are available on the platforms listed above. However, you should always check the WebSphere Message Broker site for the latest updates.

Prerequisite products

- z/OS
 - ▶ z/OS V1.5 or later
 - ▶ WebSphere MQ V5.3.1 or later
 - ▶ DB2® V7.1 (+PTFs) or later
 - ▶ Java™ V1.4.2
 - ▶ XML Toolkit v1.8
- Distributed
 - ▶ JRE V1.4.2
 - ▶ WebSphere MQ V5.3 or V6.0
 - ▶ Data Base (DB2 V8.2, Oracle or Sybase or SQL Server)
- Workbench/Toolkit (in addition to other distributed prerequisites)
 - ▶ Eclipse Platform V3.0.1
 - ▶ IBM Rational® Agent Controller V6.0 (for debugger)



The pre-requisites for the Broker are listed above for z/OS and the distributed platforms like UNIX®, and Windows®. The Systems Administration and Application Development component, and the Broker Toolkit, which runs on either Windows or Linux, run on the Eclipse Platform version 3.0.1; the Rational Agent Controller is required if you use the debugger.

Section

WebSphere Message Broker components



This section explains the components of the Message Broker and their functions.

WebSphere Message Broker and WebSphere Event Broker components

The main components of WebSphere Message Broker and WebSphere Event Broker are:

- Workbench/Toolkit (application development)
- Configuration Manager (system administration)
- Broker (runtime)
- User Name Server – optional pub sub lists



The Broker products are comprised of four components.

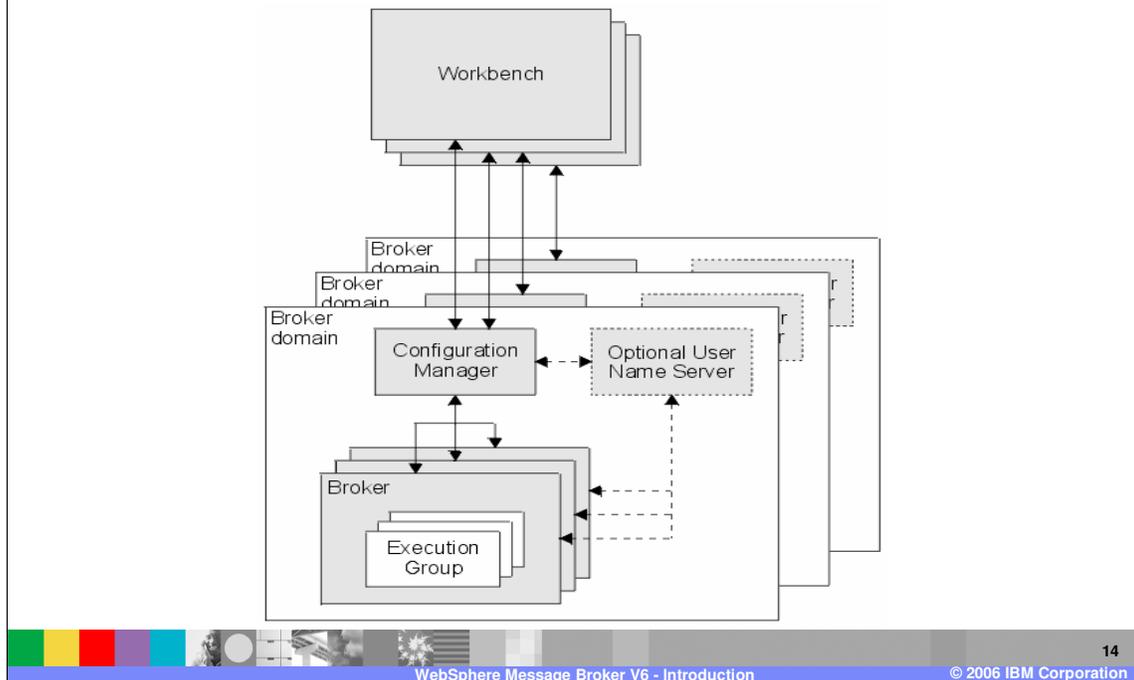
The Toolkit or Workbench is an Eclipse based GUI interface which provides a workspace for application development; systems administration functions may also be performed from the Toolkit.

The Configuration Manager is the component that allows you to administer your broker environment.

The Broker is the runtime component, where the actual work of routing and transforming messages takes place.

The User Name Server is an optional component that provides a method of maintaining publication and subscription lists.

Components



The workbench or toolkit runs on either Windows or Linux. Communications to the Configuration Manager, which is a process that may run on any of the platforms, are provided by WebSphere MQ. The Configuration Manager manages its broker domain, providing an interface between the application development workbench and the Broker runtime.

Toolkit (workbench)

- The Message Brokers Toolkit is the development environment and systems administration component
 - ▶ based on the Eclipse platform.
- The Message Brokers Toolkit can be installed on Windows and Linux (x86 platform).
- Application development perspective
 - ▶ Message flow development
 - ▶ Message flow repository with version control
- Broker administration perspective
 - ▶ Create broker domains
 - ▶ Manage broker domains
 - ▶ Deploy message flows to brokers

The Message Brokers Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform. The Message Brokers Toolkit can be installed on Windows and Linux (x86 platform).

Application developers work in separate instances of the Message Brokers Toolkit to develop message flows.

Different perspectives in the workbench are used to develop message flows and 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.

The Message Brokers Toolkit also communicates with one or more Configuration Managers, and is used to create and manage broker domains.

Configuration manager

The Configuration Manager has four main functions:

1. Records configuration details
2. Deploys the broker topology and message processing operations to the Broker component
3. Provides status of the broker domain
4. Communicates with other components in the broker domain



The Configuration Manager has four main functions.

It maintains configuration details in an internal repository. This internal repository provides a central record of the broker domain components.

It deploys the broker topology and message processing operations in response to actions initiated through the workbench. Broker archive files, or .bar files, are deployed through the Configuration Manager to the execution groups within a broker.

It reports on the results of deployment and the status of the broker.

It also communicates with other components in the broker domain using WebSphere MQ transport services.

Configuration manager (cont.)

Each broker domain must have a unique Configuration Manager.

The Configuration Manager runtime component uses the following resources (created when the Configuration Manager is created):

- An internal repository.
- A WebSphere MQ queue manager
 - ▶ may use an existing queue manager
 - ▶ must be on the same physical system as the Configuration Manager
- A set of fixed-name queues, defined to the queue manager that hosts the Configuration Manager.



You must install, create, and start a Configuration Manager for each broker domain. Multiple Configuration Managers may exist on one or multiple machines. The Configuration Manager name must be unique.

When you create the Configuration Manager runtime component, an internal repository is also created. A WebSphere MQ queue manager is created (except on z/OS), unless one already exists; this queue manager must reside on the same physical system as the Configuration Manager. The Configuration Manager queue manager can be shared with a single broker or user name server, or both.

Configuration Manager creation also creates a set of fixed-name queues, defined to the queue manager that hosts the Configuration Manager.

Brokers

The broker runtime component:

- Is a set of running processes.
- Hosts one or more message flows to route, transform, and enrich in-flight messages.

Applications use WebSphere MQ to send messages to the broker. JMS and HTTP are also supported transport mechanisms.

The broker routes and transforms each message.

The broker:

- Uses WebSphere MQ to communicate with the Configuration Manager and other brokers in the broker domain.
- Requires a broker database to hold broker control information.
- Uses ODBC to communicate with the database.



The broker is the runtime component of WebSphere Message Broker. A broker is a set of running processes that hosts one or more message flows to route, transform, and enrich in-flight messages.

Applications send messages to the broker using WebSphere MQ queues and connections; JMS and HTTP are also supported transport mechanisms. 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 WebSphere MQ sender and receiver channels to communicate with the Configuration Manager and other brokers in the broker domain.

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

The broker connects to the database using an ODBC connection.

Broker domains

A broker domain is:

- One or more brokers that share a common configuration
- Controlled by a single Configuration Manager

A broker name must be unique within the broker domain, the same name used in the broker domain topology in the workbench. Information is kept in the configuration repository.

Create a broker reference and deploy the changes to your broker domain.

Deployment:

- starts communications between the broker and the Configuration Manager.
- initializes the broker to make it ready to handle message flows.



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

A broker name must be unique within the broker domain. This is the same name used 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 handle message flows.

Broker creation

When you create a broker, the following resources are also defined and created:

- Database tables for storing the information used by the broker
- A WebSphere MQ queue manager (except on z/OS), unique for each broker
- A set of fixed-name queues within the WebSphere MQ queue manager.



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.
- A WebSphere MQ queue manager (unique for each broker), if one does not already exist (except on z/OS).
- A set of fixed-name queues that are defined to the WebSphere MQ queue manager.

User Name Server

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

User Name Server requires

- A WebSphere MQ queue manager.
- A set of fixed-name queues, defined to the WebSphere MQ queue manager that hosts the User Name Server.

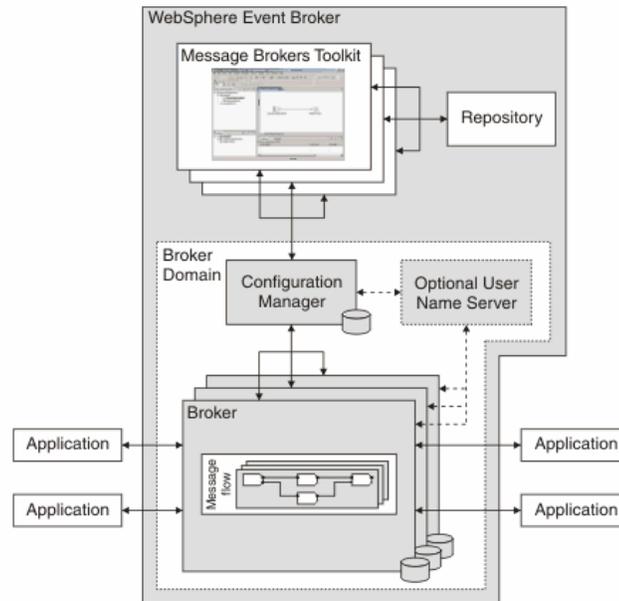


The User Name Server is an optional runtime component that can provide authentication of users and groups performing publish/subscribe operations.

When you create a User Name Server, the following resources are also created:

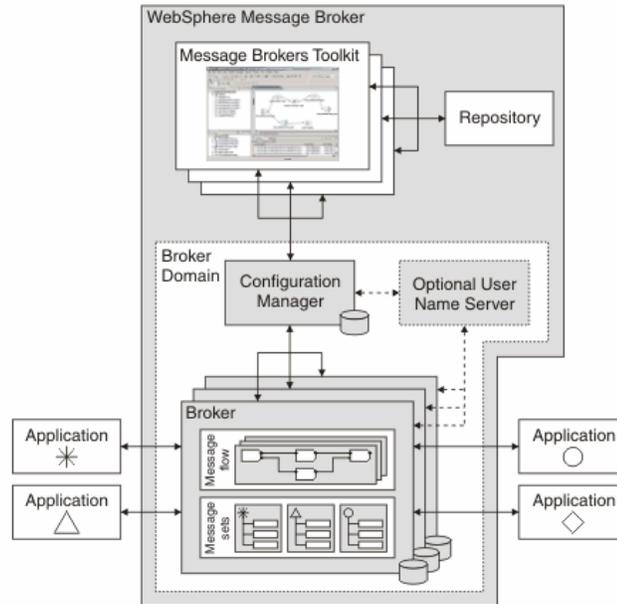
- A WebSphere MQ queue manager unless one exists already (except on z/OS). The queue manager is created on the same physical system as the User Name Server. The user name server queue manager can be shared with the Configuration Manager, a single broker, or both.
- A set of fixed-name queues, defined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

WebSphere Event Broker



In the WebSphere Event Broker, publishing applications send data to the message flow. The message flow sends out the information to subscribers, as dictated by the subscribers' topics and authorizations.

WebSphere Message Broker



In the WebSphere Message Broker, many applications interface with their message flows, not just the publish and subscribe applications. The message flows may modify, transform and reroute a message as dictated within the flow. The message flow manipulation is transparent to both the sending application and the receiving application. This provides a method for legacy applications to communicate without change to new applications. It also provides a single point for additions and changes to the environment.

Section

Summary and references

The last portion of the presentation contains a summary and references.

Summary

- WebSphere Brokers provide a way to integrate your business processes and applications.
- The Event Broker provides a publish/subscribe capability.
- The Message Broker provides, in addition, the capability to reroute and transform messages.



WebSphere Brokers help you integrate your business processes and applications by bringing together the data and information of your enterprise, allowing you to achieve business goals.

The broker sits between applications and processes each message independent of the applications. No change to applications is necessary. It is an ideal way to connect legacy applications with new technologies such as web-based interfaces.

The Event Broker provides a publish/subscribe model that allows distribution of real-time information from disparate sources through a centralized broker.

The Message Broker provides the framework supporting basic functions and user-defined enhancements which facilitate rapid development and modification of business processing rules that are applied to the messages in the system.

References

- WebSphere Message Broker library:

<http://www-306.ibm.com/software/integration/wbimessagebroker/library/>

- WebSphere Message Broker Information Center:

<http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp>

- WebSphere Message Broker home page:

<http://www-306.ibm.com/software/integration/wbimessagebroker/>

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e/Logo/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2005,2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.