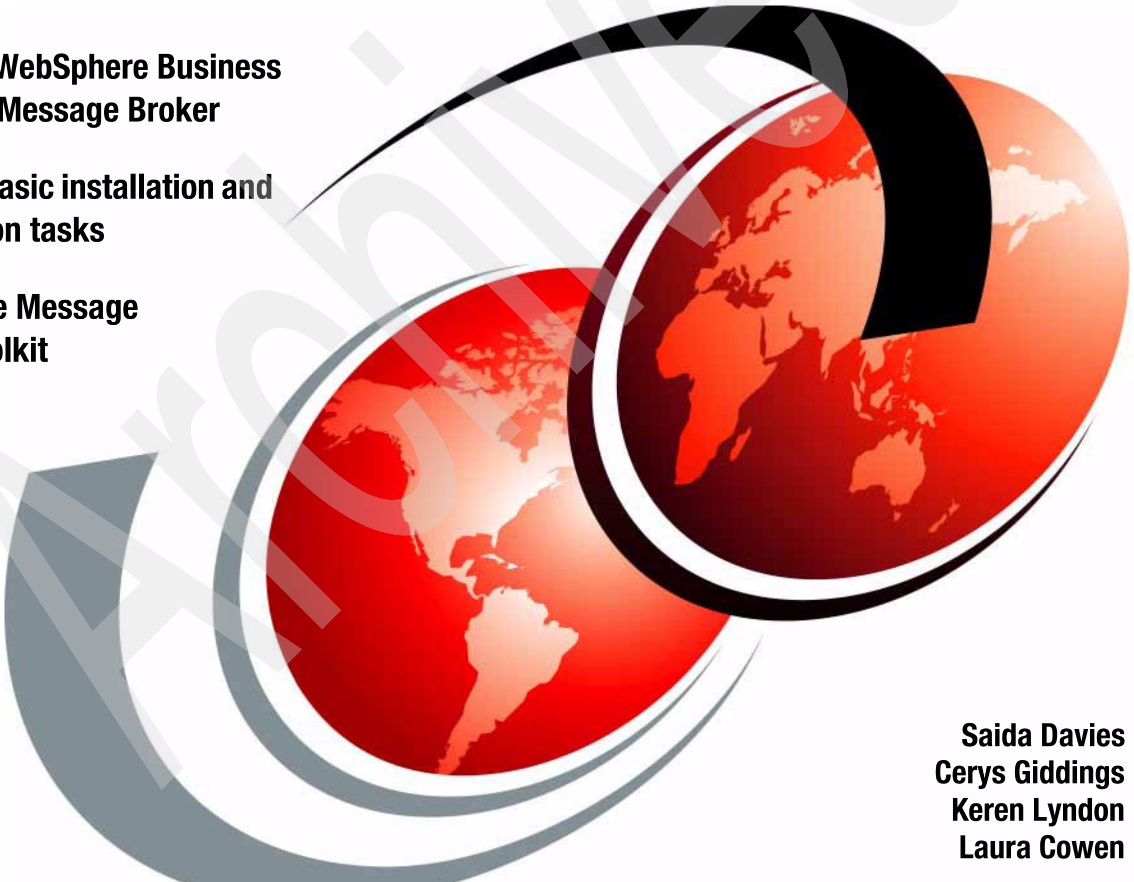
IBM

# WebSphere Business Integration Message Broker Basics

Introduces WebSphere Business Integration Message Broker

Describes basic installation and configuration tasks

Explores the Message Brokers Toolkit



Saida Davies
Cerys Giddings
Keren Lyndon
Laura Cowen

Redbooks

ibm.com/redbooks

International Technical Support Organization

# WebSphere Business Integration Message Broker Basics

June 2004

SG24-7090-00

**First Edition (June 2004)**

This edition applies to Version 5.0.2 of WebSphere Business Integration Message Broker and WebSphere Business Integration Event Broker.

# Contents

# Notices

# Trademarks

# Preface

This IBM® Redbook is a guide to the installation and basic configuration of WebSphere® Business Integration Message Broker V5.0 and WebSphere Business Integration Event Broker V5.0. It introduces message flow applications development, and instructions for creating a simple message flow and message set are provided.

We also describe how to configure broker domains and how to deploy message flow applications to the broker.

The book includes details about where to look for diagnostic information in the product and where to find more information about the product, including product documentation updates and sample applications.

This redbook is intended for users who need to get up to speed with using the new version of the product and the Message Brokers Toolkit.

## The team that wrote this redbook

This redbook was produced by a team of specialists from the home of Business Integration and Messaging Middleware product development, IBM Hursley.

From left: Cerys, Keren, and Laura
Below: Saida

**Saida Davies** is a Project Leader at International Technical Support Organization (ITSO); she is a certified Senior IT Specialist and has 15 years of experience in IT. Saida has published several Redbooks™ about various business integration scenarios. She has experience in the architecture and design of WebSphere MQ solutions, extensive knowledge of IBM z/OS®, and a detailed working knowledge of both IBM and Independent Software Vendors' operating system software. In a customer-facing role with IBM Global Services, her role included the development of services for WebSphere MQ within z/OS and Windows®. This covered the architecture, scope, design, project management, and implementation of the software on stand-alone systems and on systems in a Parallel Sysplex® environment. She has a degree in Computer Science and her background includes z/OS systems programming.

**Cerys Giddings** is a Software Engineer and has worked on the WebSphere MQ family of products since joining IBM Hursley in 2000. She has participated in producing the IBM Certified System Administrator and WebSphere Business Integration Message Broker V5 certification processes. She has 10 years of experience in providing IT education and support, and holds a degree and Masters from the University of Wales, as well as the BCS Professional Examination at Certificate and Diploma levels.

**Keren Lyndon** is a usability practitioner and has worked with the IBM WebSphere family of products since joining IBM Hursley, United Kingdom, as a graduate in 2000. Keren is an advocate of user-centered design and very keen to produce this redbook in order to enhance the initial experience for users of the WebSphere Business Integration Broker products. She has seven years of experience in the Information Technology industry, including time spent in Frankfurt, Germany, as a Web information developer. Keren holds a degree in modern languages and information technology from the University of the West of England in Bristol, United Kingdom.

**Laura Cowen** is an Information Developer who has worked on WebSphere Business Integration Message Broker and WebSphere MQ since joining IBM Hursley in 2001. She specializes in writing documentation to go alongside enhancements to the user experience of products in the WebSphere MQ family, including sample applications for WebSphere Business Integration Message Broker. While working on WebSphere Business Integration Message Broker, she gained expertise in delivering Eclipse-based help systems. Prior to joining IBM, Laura worked as a usability researcher. She is now the editor of a quarterly magazine for the British Computer Society. She holds a degree and a Masters in human-computer interaction from Lancaster University.

The redbook team thanks the following people for their guidance, assistance, and contributions to this book:

Kuldeep Bhardwaj, Software Engineer, IBM

Chris Billing, Product Usability Design Administrator, IBM

Neil Dewhurst, Information Developer for WebSphere MQ, IBM Hursley

Matt Lucas, WebSphere Business Integration Brokers Development, IBM Hursley

Melita Saville, Platform Support Manager, IBM Hursley

Kavitha Suresh Kumar, Senior Software Engineer, IBM Global Services India

James Taylor, WebSphere Business Integration Brokers Development, IBM Hursley

Betsy Thaggard, ITSO Editor, IBM Austin

## Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

**ibm.com**/redbooks/residencies.html

## Comments welcome

Your comments are important to us. We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  **ibm.com**/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

# Introduction and overview

This chapter describes the scope of this book and introduces WebSphere Business Integration Message Broker, WebSphere Business Integration Event Broker, and their main components.

This chapter covers:

- The scope of this redbook
- WebSphere Business Integration Event Broker
- WebSphere Business Integration Message Broker
- The main Event Broker and Message Broker components
- Message Brokers Toolkit for WebSphere Studio

## 1.1 The scope of this redbook

The aim of this book is to introduce new users to the concepts and basic tasks of using WebSphere Business Integration Message Brokers on Windows platforms.

The following WebSphere Business Integration products are introduced:

- WebSphere Business Integration Event Broker V5.0
- WebSphere Business Integration Message Broker V5.0

WebSphere Business Integration Event Broker contains a subset of the features that are available in WebSphere Business Integration Message Broker. Therefore, this book discusses the features that are available in Message Broker and highlights when a feature is not available in Event Broker (for example, message sets).

This book is also relevant to users who are migrating from previous versions of the product and who need to become familiar with the Message Brokers Toolkit for WebSphere Studio (the graphical user interface for developing WebSphere Business Integration Message Brokers applications). However, other sources of information cover product migration in detail, so the actual migration of a system or configuration is covered only at a high level.

As well as installing and configuring WebSphere Business Integration Message Brokers, this book also discusses how to develop and deploy simple message flow applications in the Message Brokers Toolkit for WebSphere Studio. An overview of where to look for error information and how to diagnose problems is also provided.

The Message Brokers Toolkit is available only on the Microsoft® Windows 2000 and Windows XP platforms; therefore this book covers the tasks that can be performed on Windows platforms.

This redbook assumes that the user is working from Version 5.0.2, which is a refresh of the product. For users of Version 5.0.0, the instructions for installation are very similar. We recommend that users of Version 5.0.0 install the latest fix pack before following any of the other instructions in this redbook. See Chapter 3, “Installation” on page 23 for information about how to install a fix pack, and see 3.3, “Post-installation tasks” on page 50 for a known issue affecting Version 5.0.0 of WebSphere Business Integration Message Brokers.

## 1.2 WebSphere Business Integration Message Brokers

Throughout this book, the use of the generic name WebSphere Business Integration Message Brokers refers to both the WebSphere Business Integration Event Broker and the WebSphere Business Integration Message Broker products.

WebSphere Business Integration Message Brokers is used to:

- Securely connect business applications with those of business partners, suppliers, and customers.
- Exchange information between applications regardless of the information format, the platforms on which the applications are installed, and the geographical locations and time zones of the applications.
- Integrate dissimilar computer systems that an organization company owns as a result of mergers and acquisitions.

The WebSphere Business Integration Message Brokers products use WebSphere MQ messaging and queuing technology to transport information between business applications in the form of messages. WebSphere MQ provides assured, once-only delivery of each message.

### 1.2.1 WebSphere Business Integration Event Broker

Event Broker provides a publish/subscribe style of messaging. In publish/subscribe messaging, the application that sends the information does not know which applications receive the information. The sender application publishes messages that contain the information and applications receive the information according to the topics that they have subscribed to.

For example, a service could provide football, cricket, and golf scores to customers while the games are being played. If a customer subscribes their mobile phone number to receive the football and cricket scores, the customer receives text messages about those particular topics as the information is published. The customer does not receive the golf scores unless they explicitly subscribe to the golf scores. Alternatively, the customer could subscribe to receive the scores from all sports events, in which case they receive the football, cricket, and golf scores.

In this example, the text messaging service publishes the information regardless of who is receiving it; only customers who have subscribed to the information receive it.

### 1.2.2 WebSphere Business Integration Message Broker

Message Broker incorporates and extends WebSphere Business Integration Event Broker. As well as the publish/subscribe style of messaging, Message Broker provides point-to-point messaging. In point-to-point messaging, the application that sends the information knows which applications receive the information. The benefit of point-to-point messaging is that the broker can route the messages according to the message content. The broker can also manipulate the message content, interact with databases, and even transform the message from one format to another.

### 1.2.3 Components of Event Broker and Message Broker

The following sections briefly describe the main components of Event Broker and Message Broker:

- Broker
- Broker domain
- Configuration Manager
- Message flows
- Message sets
- Execution group
- User Name Server

#### Broker

The broker is the component that represents WebSphere Business Integration Message Brokers at runtime. When a message arrives at the broker from a business application, the broker processes the message before passing it on to one or more other business applications. The broker can route, transform, and manipulate messages according to the logic that is defined in its message flows. Each broker has a database in which it stores the information that it needs to process messages at runtime.

The system administrator can create and configure many brokers on many different platforms, including Unix and z/OS systems. For example, configure multiple brokers as backups in case one broker fails, and create separate brokers for separate company divisions or geographical locations.

#### Broker domain

Brokers are grouped together in broker domains. The brokers in a single broker domain share a common configuration that is defined in the Configuration Manager. You can create one or more broker domains on a system, but each broker domain must have its own Configuration Manager.

## Configuration Manager

The Configuration Manager manages the broker domain and is the interface between the Message Brokers Toolkit and the brokers in the broker domain. It stores information about the configuration of a broker domain in a database. The Configuration Manager is responsible for deploying message flow applications to the broker and reports back on the progress of the deployment and on the status of the broker. This feedback is displayed in messages. To use the Message Brokers Toolkit, a Configuration Manager must have been created and started. A Configuration Manager can be created only on Windows platforms.

Chapter 5, “Administration” on page 133 describes how to create and configure brokers and a Configuration Manager in a broker domain.

## Message flows

Message flows are programs that provide the logic that the broker uses to process messages from business applications. Message flows are created in the Message Brokers Toolkit using the graphical Message Flow Editor to click and place nodes and connect them together. Each node performs some basic logic, and Extended Structured Query Language (ESQL) can be used to program many of the available nodes to perform complex manipulations and transformations of the messages. This includes accessing databases and creating new messages.

Figure 1-1 shows a simple message flow that is used in Chapter 4, “Application development” on page 83. Message flows on a production system are typically much more complex with many more nodes. All message flows comprise a collection of nodes that are connected to determine the order in which the nodes process the message. The content of the message itself can also determine the route that it takes through a message flow.

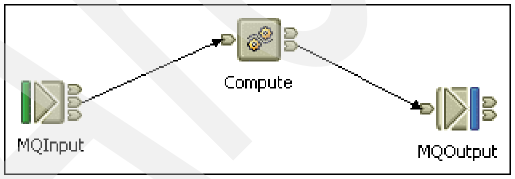


*Figure 1-1 An example of a very simple message flow*

## Message sets

A message set is a template for the structure of the messages that are processed by the message flows in the broker. The message flow refers to the message set to parse the messages that the broker receives from business applications. The message flow can also use a message set to map the fields of an input message onto the fields of an output message to create a partial copy of the input message.

Message flows can parse and perform limited manipulations on messages without a message set if the messages are self-defining and contain all of the information about their structure within the message content. For example, Extensible Markup Language (XML) messages define the fields using XML tags, which the message flow can parse. ESQL can then be used to program the message flow to manipulate the message. However, if the manipulations are complex, it is often easier to predefine the message structures in a message set and, subsequently, reduce the amount of programming and debugging.

**Note:** Message sets are not available in WebSphere Business Integration Event Broker.

### Execution group

Execution groups enable message flows within the broker to be grouped together. Each broker contains a default execution group, and more execution groups can be created as long as unique names are given to them within the broker. Each execution group is a separate operating system process so that the contents of an execution group remain separate from the contents of another execution group within the same broker, which can be useful for isolating pieces of information for security.

See Chapter 5, “Administration” on page 133 for more information about execution groups.

### User Name Server

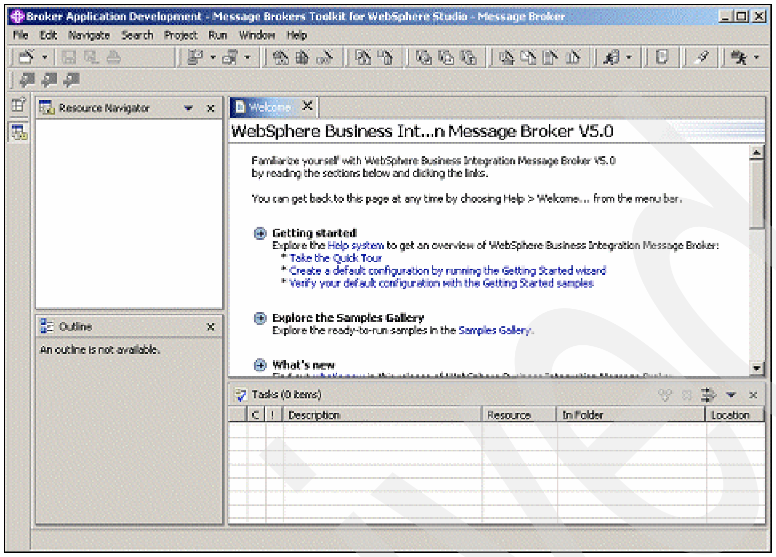
A User Name Server is an optional component that is needed only if publish/subscribe applications are run. A User Name Server authenticates users and groups that are publishing and subscribing to particular topics of information.

See Chapter 5, “Administration” on page 133 for more information about User Name Servers.

## 1.3 Message Brokers Toolkit for WebSphere Studio

The Message Brokers Toolkit for WebSphere Studio, or Message Brokers Toolkit, is the graphical user interface for the WebSphere Business Integration Message Brokers V5.0 products. (See Figure 1-2 on page 7.) The Message Brokers Toolkit replaces the Control Center from previous versions of WebSphere Business

Integration Message Brokers. The Message Brokers Toolkit currently runs only on Windows 2000 and Windows XP.


Broker Application Development - Message Brokers Toolkit for WebSphere Studio - Message Broker
File Edit Navigate Search Project Run Window Help
Resource Navigator
Outline
Welcome

An outline is not available.

WebSphere Business Int...n Message Broker V5.0

Familiarize yourself with WebSphere Business Integration Message Broker V5.0 by reading the sections below and clicking the links.

You can get back to this page at any time by choosing Help > Welcome... from the menu bar.

**Getting started**
Explore the Help system to get an overview of WebSphere Business Integration Message Broker:
* Take the Quick Tour
* Create a default configuration by running the Getting Started wizard
* Verify your default configuration with the Getting Started samples

**Explore the Samples Gallery**
Explore the ready-to-run samples in the Samples Gallery.

**What's new**

Tasks (0 items)

| | C | ! | Description | Resource | In Folder | Location |
|---|---|---|---|---|---|---|

*Figure 1-2 Message Brokers Toolkit for WebSphere Studio*

The main purpose of the Message Brokers Toolkit is for developers to develop, test, and deploy message flow applications. Although some administrative tasks can be performed in the Message Brokers Toolkit, such as deploying message flow applications and viewing the topology of a broker domain, tasks such as creating a broker or creating a Configuration Manager must be performed from the command line.

To use the Message Brokers Toolkit, it must be installed on a Windows 2000 or Windows XP system, even if connecting to a Configuration Manager or broker on a remote system.

The Message Brokers Toolkit comprises a collection of perspectives. A perspective (the full workbench window) is a collection of views that can be moved and re-sized. Figure 1-2 shows the Broker Application Development perspective of WebSphere Business Integration Message Broker, which includes the Resource Navigator view (upper left), the Outline view (lower left), the Editor view (upper right), and the Tasks view (lower right).

The first time that the Message Brokers Toolkit is opened, the Broker Application Development perspective is displayed. To open a new perspective:

1. Click **Window** → **Open Perspective**.
2. From the list, click the name of the perspective that is to be used.

When a new perspective has been opened, a button for the perspective is added to the toolbar on the left side of the Message Brokers Toolkit window. Figure 1-3 shows the buttons for the Broker Application Development perspective, the Broker Administration perspective, and the Flow Debug perspective. In the figure, the Flow Debug perspective is currently in use.

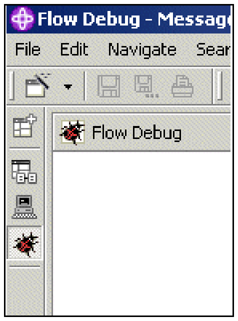


*Figure 1-3 Perspective toolbar buttons*

Table 1-1 shows which perspective is needed for each of the most common tasks that are performed in the Message Brokers Toolkit.

*Table 1-1 Perspectives to use for common tasks*

| **Task** | **Perspective** |
|---|---|
| Developing message flows | Broker Application Development perspective |
| Developing message sets | Broker Application Development perspective |
| Deploying and testing message flow applications | Broker Administration perspective |
| Debugging message flow applications | Flow Debug perspective and Debug perspective |
| Managing broker domains | Broker Administration perspective |
| Configuring publish/subscribe applications | Broker Administration perspective |

The information that is visible within a perspective depends on the tasks that you are performing. However, you can fully customize the layout and content of each perspective. You can also reset a perspective to its default layout at any time. To find out more about the Message Brokers Toolkit, see the *Workbench User Guide* and the product documentation: **Reference** → **Workbench**.

2

# Planning and security considerations

This chapter provides details of software and system considerations for using WebSphere Business Integration Message Brokers. Security issues relating to the product and installation are discussed. A brief overview of migration tasks for WebSphere MQ Integrator V2.1 is also given.

The following topics are discussed:

- System requirements
- Software resources
- Security issues
- Migration tasks

## 2.1 System requirements

This section details the system requirements for the WebSphere Business Integration Message Brokers product. This includes discussion of hardware considerations, supported operating systems, and other system-related requirements.

### 2.1.1 Supported operating systems

The information provided in this section is a brief overview of the supported operating systems and does not cover fix packs, maintenance levels, or other such patch levels, as these change with time and over the product life cycle. The latest relevant information about these can be obtained from the product README, which is available on the WebSphere MQ family Web site. See the Links section in Appendix A, "Getting help" on page 219.

Not all of the components in WebSphere Business Integration Message Brokers can be installed onto each of the supported operating systems. Table 2-1 shows which WebSphere Business Integration Message Brokers components are supported on various operating systems.

*Table 2-1 WebSphere Business Integration Message Brokers operating system support*

| Supported operating system | Message Brokers Toolkit | Configuration Manager | Broker | User Name Server |
|---|---|---|---|---|
| AIX® | n/a | n/a | Supported | Supported |
| HP-UX | n/a | n/a | Supported | Supported |
| Solaris | n/a | n/a | Supported | Supported |
| Linux Intel® | n/a | n/a | Supported | Supported |
| Linux on zSeries® | n/a | n/a | Supported | Supported |
| z/OS | n/a | n/a | Supported | Supported |
| Windows 2000 Professional | Supported | Supported | Supported only for development and test (not production) | Supported |
| Windows 2000 Server | Supported | Supported | Supported | Supported |

| Supported operating system | Message Brokers Toolkit | Configuration Manager | Broker | User Name Server |
|---|---|---|---|---|
| Windows 2000 Advanced Server | Supported | Supported | Supported | Supported |
| Windows XP Professional | Supported | Supported | Supported only for development and test (not production) | Supported |
| Windows 2003 Standard | Not supported | Supported | Supported | Supported |
| Windows 2003 Enterprise | Not supported | Supported | Supported | Supported |

As the table shows, it is necessary to use a machine that is running the Windows operating system to perform message application development and deployment with the Message Brokers Toolkit. However, the other operating systems are most suitable for production broker runtime systems with the tested message applications deployed to them.

### 2.1.2 Hardware considerations

The hardware onto which the WebSphere Business Integration Message Brokers products are installed is largely dependent on the operating systems being used in the configuration. Table 2-2 shows a list of supported hardware and associated operating systems. For the most up-to-date information, check the latest README on the WebSphere MQ family Web site.

*Table 2-2 WebSphere Business Integration Message Brokers hardware support*

| Supported operating system | Related hardware |
|---|---|
| AIX | IBM pSeries®<br>IBM RS/6000® processor machines |
| HP-UX | Hewlett-Packard HP-9000 processor machines |
| Solaris | Sun Microsystems SPARC processor machines |
| Linux Intel | IBM xSeries® or equivalent Intel-based systems |

| Supported operating system | Related hardware |
|---|---|
| Linux on zSeries | IBM zSeries 600 or 700 server |
| z/OS | Any server capable of running IBM z/OS |
| Windows | IBM xSeries or equivalent Intel-based system<br>IBM iSeries™ Server using the IBM Integrated xSeries Server |

## 2.1.3 Disk space

Before installing the WebSphere Business Integration Message Brokers product, consideration must be given to the amount of disk space that is required on the system. This not only includes the amount of disk space that WebSphere Business Integration Message Brokers takes up after installation, but also space for temporary files during install, space for the installation of prerequisite software, and space for post-installation configuration and development.

### Disk space for the product

Installation of the WebSphere Business Integration Message Brokers runtime components requires approximately 365 MB for a typical install of the Event Broker product and 425 MB for the Message Brokers product.

The Message Brokers Toolkit requires approximately 400 MB of disk space.

The total space required for the installed product is likely to increase by the order of tens to hundreds of megabytes as resources are developed, logs are generated, and service is applied. This increase in space requirement over time should be considered, but there are mechanisms to control the space requirements, such as cleaning up logs and archiving resources. One of the main sources of additional space usage is the application of service when the Message Brokers Toolkit is installed. The reason for this is that new levels of plug-ins for the Message Brokers Toolkit may be installed with service, and therefore many levels of the same plug-in may be present on a system.

Some additional space may be required for temporary files during the install. Any temporary files are deleted when the install is complete.

### Database disk space

DB2® is the required database for the Configuration Manager component, but for the broker the database can be DB2, Oracle, Sybase, or SQL Server on Windows. See the product documentation for more information.

A DB2 installation may require between 350 MB and 500 MB of disk space. If an additional database is installed on the system (for example, Oracle or Sybase to be used for the broker database or user databases) then significantly more disk space is required. A typical Sybase installation is around 700 MB and a typical Oracle installation is more than 4.5 GB.

In addition to the installation disk space for databases, extra disk storage is needed to for the configuration and running of databases for the broker repository, the Configuration Manager respository, and user databases. For the broker and Configuration Manager repository, the minimum disk space that is required for each database is 10 MB.

#### WebSphere MQ disk space

WebSphere MQ requires approximately 150 MB of disk space to install, but the required disk space increases with the objects that are created on the system and the settings of any logging that is set up on the system. Persistent message logging requires more disk space than non-persistent messaging, for example.

#### Message Brokers Toolkit Workspace disk space

Resources such as message flows and message sets that are created in the Message Brokers Toolkit are stored on the local file system in a workspace directory. This is also known as the Eclipse workspace. The size of the workspace directory is likely to increase over time and depends on the number and types of resources.

### 2.1.4 System memory

The amount of system memory that is required for running the WebSphere Business Integration Message Brokers runtime is 512 MB. The minimum amount of system memory for the Message Brokers Toolkit is 512 MB, although it may run slowly, and 1 GB of RAM would be the recommended amount.

System memory that is required for the prerequisite software also must be given consideration.

## 2.2 Software resources

This section discusses the prerequisite software for WebSphere Business Integration Message Brokers. However, it is possible to perform some tasks in WebSphere Business Integration Message Brokers without having all of the prerequisite software, so we describe the function of each program and what tasks can and cannot be performed without it. Other products that are supported for use with the WebSphere Business Integration Message Brokers product are also covered.

Operating system patches, internal components, and Java™ are not discussed in this section.

### 2.2.1 IBM DB2 Universal Database Enterprise Server Edition

In WebSphere Business Integration Message Brokers V5.0, both Version 7.2 and Version 8.1 of IBM DB2 Universal Database™ Enterprise Server Edition (DB2) are supported. Refer to the latest online documentation for the required fix pack level for the version of DB2 that is to be used with the product. DB2 V8.1 is supplied with WebSphere Business Integration Message Brokers for use with the broker and Configuration Manager.

DB2 is a prerequisite for the Configuration Manager because it is used as the database storage for the configuration repository. DB2 can also be used as the broker database, although other database products may also be used as the broker database, as discussed in 2.2.4, "Supported broker databases" on page 15. However, DB2 must be used for the Configuration Manager repository.

If DB2 is not installed on a system with WebSphere Business Integration Message Brokers, then the Configuration Manager cannot be created on the system. However, brokers could be created if another supported database is installed on the system.

The Message Brokers Toolkit can connect to a Configuration Manager on a remote machine if required, so development of broker applications can be carried out without DB2 being installed on the system. Database mappings can be created using a database on a remote system if the Open Database Connectivity (ODBC) connections to that database have been set up correctly.

Most tasks in the Message Brokers Toolkit can be performed without DB2 on the system; the major exceptions to this are using the Getting Started Wizard to create a default configuration and running the sample resource creation wizards. These expect DB2 to be installed on the local machine, and they use only DB2 to create Configuration Manager, broker database, and user database tables.

If DB2 is used as the broker database, then the database connection must be defined using ODBC. The DB2 database that is used for the Configuration Manager repository does not have be defined in ODBC, as Java database connectivity (JDBC) is used as for the connection to this database.

DB2 V8.1 is supplied with WebSphere Business Integration Message Brokers.

### 2.2.2 IBM WebSphere MQ

IBM WebSphere MQ V5.3 is a prerequisite for the Configuration Manager, User Name Server, brokers, and communication between the Message Brokers Toolkit and the Configuration Manager. Refer to the latest online documentation for the required fix pack level for WebSphere MQ.

A separate WebSphere MQ queue manager is required for each broker that is created on a system. The Configuration Manager and User Name Server also require a queue manager, but both can share a queue manager with the same broker.

WebSphere MQ must be installed on the local system for the Message Brokers Toolkit to communicate with the Configuration Manager. Without WebSphere MQ, only development tasks can be carried out in the Message Brokers Toolkit. No default configuration can be set up or deployed, even where a remote configuration may be set up.

IBM WebSphere MQ V5.3 is supplied with the WebSphere Business Integration Message Brokers product.

### 2.2.3 IBM Agent Controller

IBM Agent Controller is used by the Message Brokers Toolkit during message flow debugging. It is also used during Rapid Application Development (RAD) deployment. Most tasks can be performed in the Message Brokers Toolkit without the IBM Agent Controller being installed on the same system, although errors may be thrown during RAD deploy. RAD deploy is discussed in upcoming chapters, and in detail in 5.2, “Deploying message flow applications” on page 147. The IBM Agent Controller must be installed onto any machine where a broker is running and where debugging may be required.

### 2.2.4 Supported broker databases

As previously mentioned, in addition to DB2, the following databases are supported as databases for holding broker configuration data:

- Sybase
- Oracle
- SQL Server

Connections to these databases must be defined in ODBC. These databases cannot be used for the Configuration Manager repository. Information about the supported versions and fix pack levels can be found in the WebSphere Business Integration Message Brokers documentation.

## 2.3 Security issues

Security is an important consideration with WebSphere Business Integration Message Brokers, and different forms of security control are available to cover different aspects of the product use. This section discusses some of these aspects briefly for Windows; consult the product documentation for further, in-depth information for Windows and other platforms.

### 2.3.1 Security for installation

There are a number of aspects to consider when deciding on a suitable user ID to install WebSphere Business Integration Message Brokers under. When installing WebSphere Business Integration Message Brokers onto a Windows machine, the user ID must be a member of the Administrators security group. A user ID can be added to the Administrators group using User accounts or Computer Management on Windows.

There are also constraints on the length of the user ID, as Windows does not support user ID lengths of greater than 12 characters. For compatibility with other platforms and possible limits on the user IDs that are supported by database software, user ID lengths of greater than eight characters are not recommended for use on systems involved in testing and production.

The Administrator user ID cannot be used with WebSphere Business Integration Message Brokers. An attempt to use Administrator with the runtime can produce authorization errors.

### 2.3.2 Post-installation security

Security for WebSphere Business Integration Message Brokers after installation is determined in one of two ways: either a user is a member of specific security groups that are defined to Windows or a user is a member of an Access Control List that is set up using command line utilities within WebSphere Business Integration Message Brokers.

The local Windows security groups that are necessary for performing tasks in WebSphere Business Integration Message Brokers are:

- Administrators
- mqm
- mqbrkrs
- mqbrasgn
- mqbrdevt
- mqbrops
- mqbrtpic

The Administrators local security group is required to install WebSphere Business Integration Message Brokers, as previously indicated, and for running commands that operate on components, such as:

- **`mqsilist`**
- **`mqsistart`**
- **`mqsichangebroker`**
- **`mqsichangetrace`**

Any users who perform tasks using IBM WebSphere MQ must be members of the mqm local user group. When components are created, started, and running, the user ID that the machine is logged onto must be a member of the mqm local security group.

Membership of the mqbrkrs local security group is required for many tasks including starting and running the runtime components, listing the existing components and their configurations, and covering all development and deployment tasks in the Message Brokers Toolkit. The remaining security groups are used to determine which users can perform which tasks relating to development and deployment in WebSphere Business Integration Message Brokers. For example, mqbrasgn enables a user to assign resources to an execution group and mqbrtpic allows a user to modify topic.

In Fix Pack 2 of WebSphere Business Integration Message Brokers, a Security wizard is available and runs at the end of the install to create the local groups, such as mqbrkrs for the user, and enable users to be added to these groups.

In Version 2.1 of WebSphere MQ Integrator Broker, these security groups were the main form of security. In Version 5.0 of WebSphere Business Integration Message Brokers, Access Control Lists (ACLs) are used to control object-level security access. Using ACLs, permissions and authorities can be granted to both individual users and security groups. Access control can be set up for:

- Topology
- Broker
- Execution Group
- Root Topic
- Subscription

The commands for viewing, creating and deleting the ACLs are:

- **`mqsicreateaclgroup`**
- **`mqsideleteaclgroup`**
- **`mqsilistaclgroup`**

Refer to the documentation in the Message Brokers Toolkit help for more information about using ACLs to control object-level security. Also, refer to the documentation for Checking Security Requirements in the Message Brokers Toolkit help for information about setting up authentication for domains and domain controllers.

### Security in the Message Brokers Toolkit

The Message Brokers Toolkit can be launched and accessed by any user in Windows, and the level of security of Broker Application Development resources is dependent on the Windows security and file permissions that are set up on a system. Any user can create, modify, and delete broker application development resources unless security is put in place, such as altering file or directory permissions or utilizing a repository for file storage that can be secured using username and password. Resources in the Message Brokers Toolkit Workspace are local to the machine, so are under local security and protection methods.

To display or change broker domain configuration information in the Message Brokers Toolkit, the user must be a member of some or all of the previously listed security groups or members of the appropriate ACLs. If an unauthorized user attempts to connect to the Configuration Manager using the Message Brokers Toolkit, the action fails with authorization errors. This prevents an unauthorized user from changing publish/subscribe information, broker domain configurations, or deploys of message sets and message flows to any brokers. More in-depth security can also be set up by using the Domain awareness flags and channel security exits, which are described in detail in the WebSphere Business Integration Message Brokers documentation.

Topic and publish/subscribe security is managed by the User Name Server component, and users and user groups are not be visible in the Message Brokers Toolkit unless a User Name Server is created on a system and in communication with the Configuration Manager.

## 2.3.3 Other security issues

Security also must be considered for the database resources that WebSphere Business Integration Message Brokers requires for the broker information and the Configuration Manager repository. Consult the documentation for DB2 and any other supported database for the broker configuration information to check any security issues, such as restrictions on user ID length. For improved security, you should ensure that any passwords are changed if the default user ID for the database product is used. For example, leaving the db2admin password as db2admin could represent a potential security risk.

## 2.4 Migration tasks

Migration to WebSphere Business Integration Message Brokers can be carried out from WebSphere MQ Integrator Broker Version 2.1 and WebSphere MQ Event Broker Version 2.1 in very similar processes. This section gives a brief overview of the main tasks involved in migrating on Windows only, as in-depth information about migration is available from other sources.

### 2.4.1 Before installing WebSphere Business Integration Message Brokers

Certain tasks must be performed before WebSphere Business Integration Message Brokers is installed and before Version 2.1 of the product is uninstalled. The exact tasks depend on what components must remain on a system.

The pre-installation tasks that must be carried out for migration are:

1. Ensure that at least CSD04 is installed for Version 2.1.
2. Check in all resources into the Version 2.1 Control Center.
3. Delete any unwanted brokers from the Control Center and remove from the runtime.
4. Stop the runtime components.
5. Back up the Configuration Manager repository database, the Message Repository Manager database, and any broker and user databases.
6. Export all required message flows from the Control Center.
7. Use the **mqsiimpexpmsgset** command to export all required message sets.
8. For brokers that are to be migrated, note the following configuration information:
   - Name of the broker.
   - Names of the execution groups and the resources deployed to them.
   - For each deployed message flow note: additional instances, commit count, commit interval, and coordinated transactions.
9. Version 2.1 brokers can coexist with Version 5.0 brokers. For any brokers not being migrated, the configuration details for these, such as topology and topics data, must be recorded.
10. Uninstall Version 2.1 but do not select to remove the data if the intention is to migrate brokers.

## 2.4.2 After installing WebSphere Business Integration Message Brokers

After installation, the following tasks can be performed to migrate message broker application resources and Version 2.1 components to WebSphere Business Integration Message Brokers:

- Migrate Version 2.1 message flows with the **mqsimigratemsgflows** command.
- Migrate Version 2.1 message sets with the **mqsimigratemsgsets** command.
- Use the **mqsimigratetables** command to migrate the Configuration Manager if this is required.
- Use the **mqsimigratetables** command to migrate the tables of each broker.
- Use Remove Deployed Children in the Domains view of the Broker Administration perspective in the Message Brokers Toolkit to force the re-creation of the default execution group.
- Re-create the previously existing execution groups and deploy migrated message flows and message sets to them.
- Set any properties on the message flow that existed previously and deploy the configuration.

## 2.5 Post-installation configuration tasks

Post-installation configuration tasks are discussed in detail in 3.3, "Post-installation tasks" on page 50, and the manual setup of a default simple configuration is covered in Chapter 5, "Administration" on page 133. The main configuration tasks that must be performed after installation are:

- Ensure that the IBM WebSphere MQ Service is running. To start this, open Windows Services from Administrative Tools in the Windows Control Panel and locate IBM MQSeries® in the list of services. Right-click on this and select Start from the context menu.
- Ensure that DB2 is running on the machine where the Configuration Manager repository is to be created. To do this on Windows, open a command line and type `db2start`.
- Ensure that the user IDs that are to be used with WebSphere Business Integration Message Brokers are members of the ACLs or that they belong to the correct security groups (or both).

- Run **`mqsisetcapacity`** and **`mqsidisplaycapacity`** to set the license units for brokers on the machine.
- Ensure that the IBM Agent Controller service is running if this is installed.

More detailed information is available in later chapters in this book.

3

# Installation

This chapter describes how to install and verify WebSphere Business Integration Message Brokers, including how to install the latest fix pack.

We cover the following tasks:

- Installing prerequisite software
- Installing the WebSphere Business Integration Message Brokers
- Installing a WebSphere Business Integration Message Brokers fix pack
- Creating a default configuration using the Getting Started wizard
- Verifying the installation

## 3.1 Planning for installation

To obtain a fully functional installation of WebSphere Business Integration Message Brokers, install the prerequisite software shown in Table 3-1. Check the latest product README file for the latest supported versions, and see Getting help for links to the latest documentation.

You can install the Message Brokers Toolkit for evaluation purposes without any of the prerequisite software and be able to connect the Message Brokers Toolkit to a remote Configuration Manager. You can also install WebSphere Business Integration Message Brokers without installing all of the prerequisite products first. However, use of the full functionality of the product and completion of all the configuration steps that are detailed in this book cannot be accomplished unless the prerequisite software is installed.

Table 3-1 lists the prerequisite software for WebSphere Business Integration Message Brokers and why it needs to be installed.

*Table 3-1 Prerequisite software for WebSphere Business Integration Message Brokers with Fix Pack 2*

| Prerequisite software | Version | What the software is for |
|---|---|---|
| Microsoft Windows 2000 | Service Pack 3 or newer | Required if running Windows 2000. The service pack is needed for security and for WebSphere Business Integration Message Brokers to function correctly. |
| Microsoft Windows XP | Service Pack 1 | Required if running Windows XP. The service pack is needed for security and for WebSphere Business Integration Message Brokers to function correctly. |
| Java Runtime Environment (JRE) | 1.3.1 | Required. Used by the IBM Agent Controller installer and WebSphere MQ. |

| Prerequisite software | Version | What the software is for |
|---|---|---|
| IBM DB2 Universal Database Enterprise Server Edition | 7.2 with FixPak 9<br>or<br>8.1 with FixPak 2 | Used to store information about the broker domain configuration and about resources deployed to the broker. |
| Microsoft Data Access Component (MDAC) | 2.7 with Service Pack 1<br>or<br>2.7 with Service Pack 1a<br>or<br>2.8 | Used for database connections and other forms of data connections, such as data sources over the network. |
| IBM WebSphere MQ | 5.3 with CSD04 or CSD05 | Facilitates messaging between the Configuration Manager, the brokers, and business applications. |
| IBM Agent Controller | 5.0.2 | Used by the message flow debugger in WebSphere Business Integration Message Brokers. |

Most of the prerequisite software products required for the Windows system are supplied in the product package. Software that is not supplied on CDs can be downloaded from the Internet. The next section describes where to find the setup files for each prerequisite software product.

### 3.1.1 Preparing for installation

This section describes the tasks that must be performed before installing WebSphere Business Integration Event Broker or WebSphere Business Integration Message Broker.

The instructions assume that the preinstall launchpad is being used, which automatically runs when the CD marked *Insert this CD first* and labeled *WebSphere Business Integration Message Broker for Windows NT, Windows 2000, and Windows XP* is inserted.

However, the use of the preinstall launchpad is not necessary. Skip the section that describes how to run the launchpad and manually complete the tasks that the launchpad performs automatically, including checking for previous installations, and installing prerequisite software.

The preinstall launchpad provides guidance for the following tasks:

1. Checks the system for previous installations of WebSphere Business Integration Message Brokers. If necessary, the launchpad prompts to perform migration tasks before continuing.
2. Checks for and installs prerequisite software.
3. Confirms the correct license.
4. Installs WebSphere Business Integration Message Brokers.

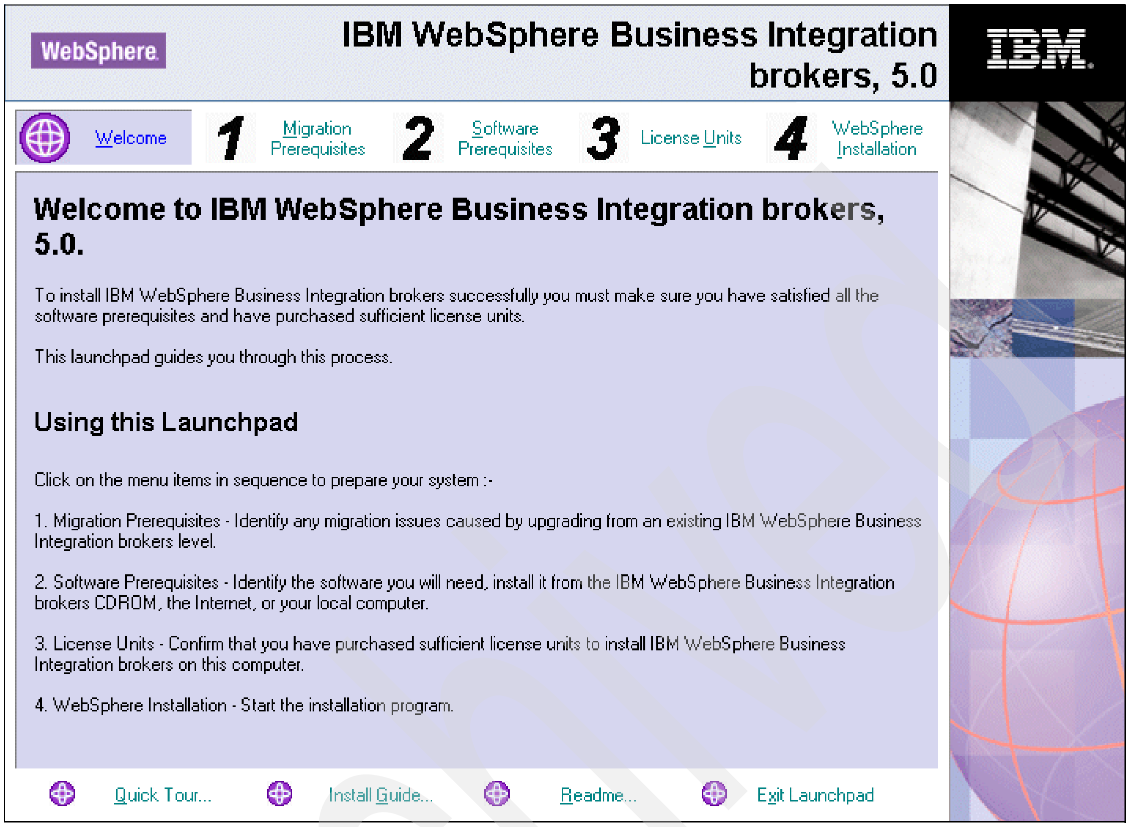
The preinstall launchpad also provides links to:

- The installation guide, which contains a link to the most recent version of the Install Guide, which is available on the Web.
- The product Readme file, which contains last minute, undocumented changes to the product and a link to the most recent version of the Readme, which is available on the Web.
- The Quick Tour, an interactive graphical overview of the product.

## Running the preinstall launchpad

To start the preinstall launchpad:

1. Locate the CD that is labeled *WebSphere Business Integration Message Broker for Windows NT, Windows 2000, and Windows XP Version 5.0.2*. This is in the product package, in the plastic wallet that contains the Windows platform CDs. The CD is also labeled *Insert this CD first*.
2. Insert the CD. The CD autoruns and the Welcome page of the preinstall launchpad is displayed (Figure 3-1 on page 27).
3. Read the most recent installation instructions:
   a. At the bottom of the Welcome page, click **Install Guide**. The Install Guide opens in a browser window.
   b. In the Install Guide, click the link to the most recent version of the Install Guide. It is essential to have a Web connection for performing this step.
4. Read the most recent release notes in the product Readme file:
   a. At the bottom of the Welcome page, click Readme. The Readme opens in a browser window.
   b. In the Readme, click the link to the most recent version of the Readme. It is essential to have a Web connection for performing this step.



IBM WebSphere Business Integration brokers, 5.0

Welcome to IBM WebSphere Business Integration brokers, 5.0.

To install IBM WebSphere Business Integration brokers successfully you must make sure you have satisfied all the software prerequisites and have purchased sufficient license units.

This launchpad guides you through this process.

Using this Launchpad

Click on the menu items in sequence to prepare your system :-

1. Migration Prerequisites - Identify any migration issues caused by upgrading from an existing IBM WebSphere Business Integration brokers level.

2. Software Prerequisites - Identify the software you will need, install it from the IBM WebSphere Business Integration brokers CDROM, the Internet, or your local computer.

3. License Units - Confirm that you have purchased sufficient license units to install IBM WebSphere Business Integration brokers on this computer.

4. WebSphere Installation - Start the installation program.

*Figure 3-1 The Welcome page*

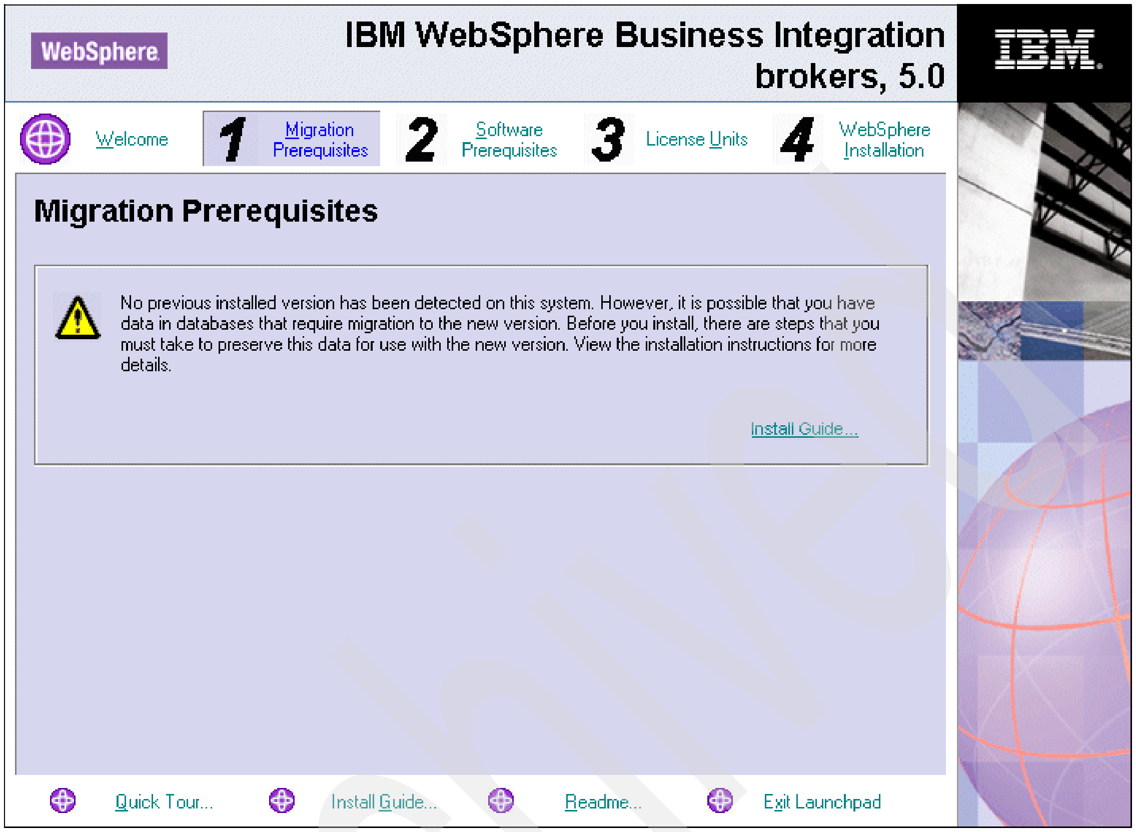
The instructions in the upcoming sections provide guidance for the steps in the preinstall launchpad.

## Checking for previous installations of the product

The Migration Prerequisites page of the launchpad checks whether the computer has a previous version of WebSphere Business Integration Message Brokers installed. A message is displayed to verify whether migration of data, message flows, and so on is required.

To check the computer for previous versions of the product:

1. At the top of the lauchpad, click **Migration Prerequisites**, which is also labeled **1**. The Migration Prerequisites page is displayed, as shown in Figure 3-2 on page 28.



Migration Prerequisites

No previous installed version has been detected on this system. However, it is possible that you have data in databases that require migration to the new version. Before you install, there are steps that you must take to preserve this data for use with the new version. View the installation instructions for more details.

*Figure 3-2 The Migration Prerequisites page*

2. The Migration Prerequisites page of the launchpad checks whether the computer has a previous version of WebSphere Business Integration Message Brokers installed.

   – If installing WebSphere Business Integration Message Brokers for the first time, and a previous version of the product is not installed on the computer, a message that no previous versions have been detected is displayed.

   – If the launchpad detects that there is a previous version of the product installed on the computer, or that there is data from a previous version of the product on the computer, then refer to the most recent version of the Install Guide for instructions about migrating the data. There is also information about migration considerations in 2.4, “Migration tasks” on page 19.

## Checking for prerequisite software

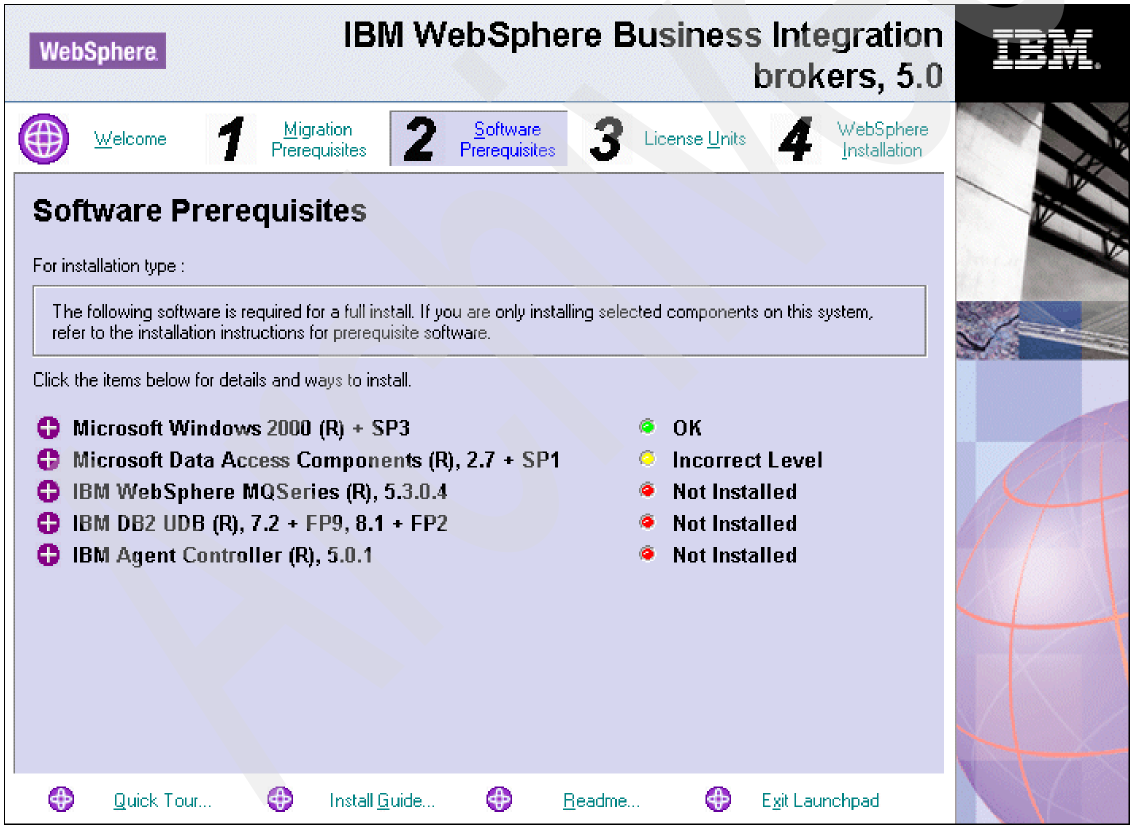
The launchpad can check for the correct versions of the prerequisite software that is installed. The launchpad indicates:

- Whether the correct version of the software is installed
- Whether the software is installed but it is the wrong version
- Whether the prerequisite software is installed at all

To check for prerequisite software:

Click **Software Prerequisites**, which is also labeled **2**. The Software Prerequisites page is displayed, as shown in Figure 3-3.

The information that is displayed on the Software Prerequisites page depends on the software prerequisites that are already installed on the computer.



Software Prerequisites

For installation type :

The following software is required for a full install. If you are only installing selected components on this system, refer to the installation instructions for prerequisite software.

Click the items below for details and ways to install.

| | |
|---|---|
| Microsoft Windows 2000 (R) + SP3 | OK |
| Microsoft Data Access Components (R), 2.7 + SP1 | Incorrect Level |
| IBM WebSphere MQSeries (R), 5.3.0.4 | Not Installed |
| IBM DB2 UDB (R), 7.2 + FP9, 8.1 + FP2 | Not Installed |
| IBM Agent Controller (R), 5.0.1 | Not Installed |

*Figure 3-3 Software Prerequisites page*

## Installing prerequisite software

To obtain a fully functional installation of WebSphere Business Integration Message Brokers, all of the prerequisite software must be installed. However, if preferred, some or all of the prerequisite software can be installed after WebSphere Business Integration Message Brokers has been installed. Refer to Table 3-1 on page 24 for details about the required software for using specific functions of WebSphere Business Integration Message Brokers.

To install a prerequisite software product:

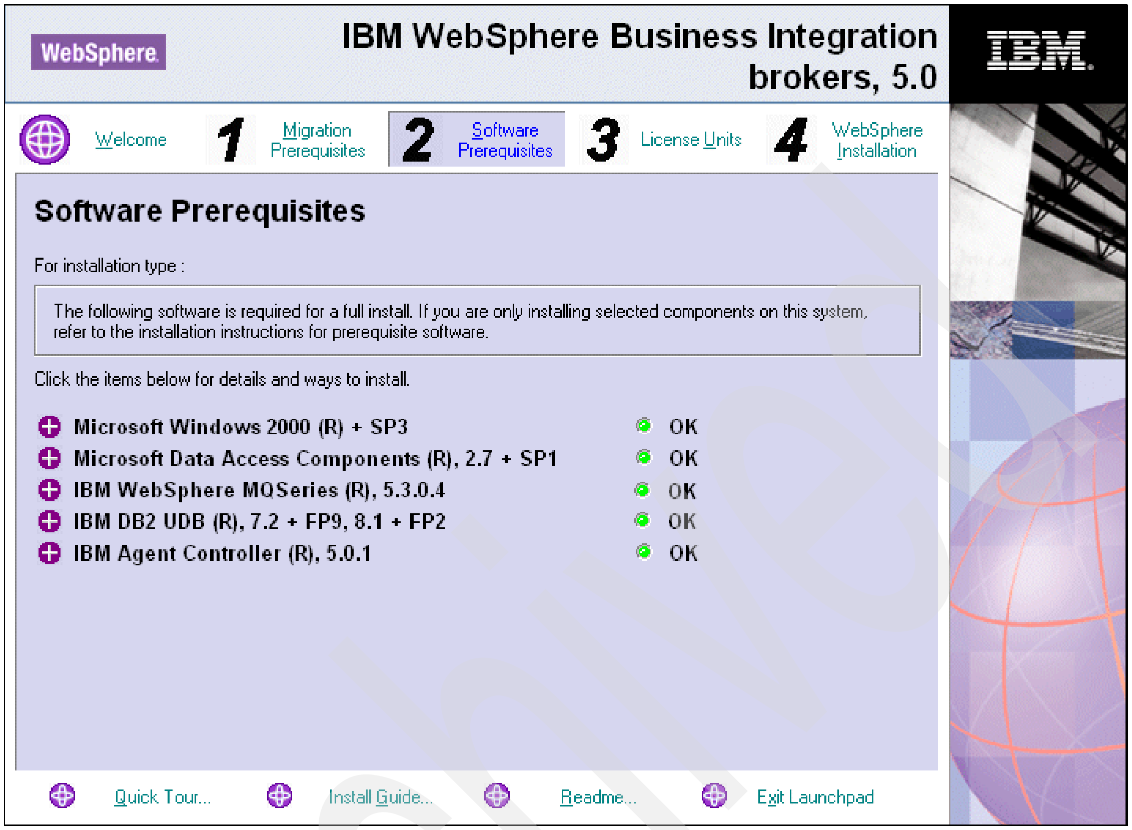
1. Click the plus sign (**+**) next to the relevant software name to expand that section of the page. This displays the installation options for the prerequisite software product.
2. Click one of the installation options:
   - CD-ROM

     Install from the CDs that are provided in the product package. This option is not available for the Microsoft Windows service packs, the WebSphere MQ CSD, or the DB2 FixPak.
   - INTERNET

     Download the setup files from the Internet and then run the installer on the computer. An Internet connection is essential for using this option. This option is not available for the IBM Agent Controller, WebSphere MQ 5.3, or DB2.
   - Network

     Install across a network. To use this option, the setup files must be available on a network drive to which access has been granted. Also ensure that the location of the setup files is mapped to the computer.

The following sections give details about where to find each prerequisite software product and how to install it. When all of the prerequisite software has been installed, the Software Prerequisites page of the launchpad reflects this, as shown in Figure 3-4 on page 31.



Software Prerequisites

For installation type :

The following software is required for a full install. If you are only installing selected components on this system, refer to the installation instructions for prerequisite software.

Click the items below for details and ways to install.

| | |
|---|---|
| Microsoft Windows 2000 (R) + SP3 | ● OK |
| Microsoft Data Access Components (R), 2.7 + SP1 | ● OK |
| IBM WebSphere MQSeries (R), 5.3.0.4 | ● OK |
| IBM DB2 UDB (R), 7.2 + FP9, 8.1 + FP2 | ● OK |
| IBM Agent Controller (R), 5.0.1 | ● OK |

*Figure 3-4 Software Prerequisites page showing successful installation of all prerequisite software*

### Installing Microsoft Windows 2000 service packs

If your computer is running Microsoft Windows 2000, Service Pack 3 or above must be installed. For computers running Windows XP, Service Pack 1 must be installed. The Service Packs fix bugs in Microsoft Windows 2000 that can prevent WebSphere Business Integration Message Brokers from functioning properly and provide greater security.

To install a Microsoft Windows service pack:

1. In the launchpad, click the **Internet** option.
2. Go to the following URL:

   `http://www.windowsupdate.com`

The service pack can be installed directly from the Windows Update site.

### *Installing Microsoft Data Access Component (MDAC) 2.7*

Upgrade the installed version of MDAC to Version 2.7 with Service Pack 1. MDAC for database connections is needed. The correct version of MDAC is supplied in the product package on CD.

To install MDAC 2.7 with Service Pack 1:

1. In the launchpad, click **CD-ROM**.
2. When prompted, insert the CD that is labeled *WebSphere Business Integration Event Broker Supplemental* or *WebSphere Business Integration Message Broker Supplemental*, then click **OK**.

   If the CD does not run automatically, use Windows Explorer to browse to the CD to the Windows\MDAC folder, and double-click **MDAC_TYP.exe** to start the installation.

### *Installing WebSphere MQ v5.3 with CSD04*

All of the messaging functionality in WebSphere Business Integration Message Brokers is provided by WebSphere MQ. Therefore, to configure WebSphere Business Integration Message Brokers, WebSphere MQ must be installed. WebSphere MQ V5.3 Server is supplied on a CD in the product package. CSD04 is not supplied in the product package; it must be downloaded from the IBM Web site and installed separately.

To install WebSphere MQ v5.3:

1. In the launchpad, click **CD-ROM**.
2. When prompted, insert the CD that is labeled *IBM WebSphere MQ for Windows Version 5.3*, then click **OK**.

   If the CD does not run automatically, use Windows Explorer to browse to the CD, and double-click **MQLaunch.exe**.

   The WebSphere MQ Installation Launchpad opens.
3. In the WebSphere MQ Installation Launchpad, start the Help Center and read the Release Notes.
4. Check that all of the WebSphere MQ software and network prerequisites are installed. Note that the WebSphere MQ Installation Launchpad cannot detect whether a Java Runtime Environment (JRE) has been installed. A JRE can be installed after WebSphere MQ has been installed.
5. In Step 3 of the WebSphere MQ Installation Launchpad, click **Launch WebSphere MQ Installer** to start installing WebSphere MQ.
6. Work through the installer and, when prompted, select the following options:
   - **Custom** installation.

   – Install the **JMS Messaging** feature.

For further instructions, refer to the WebSphere MQ product documentation and Readme.

To install WebSphere MQ V5.3 CSD04:

1. Stop all WebSphere MQ services:
   a. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**. The Services dialog opens.
   b. Right-click **IBM MQSeries**, then click **Stop**. A progress bar is displayed while the service stops.
2. In a Web browser, go to the following URL:

   `http://www-306.ibm.com/software/integration/mqfamily/support/summary/index.html`

   This is the IBM Web site from which WebSphere MQ CSDs (also known as fix packs) can be downloaded.
3. In the first table on the Web page, which is labeled IBM WebSphere MQ (formerly IBM MQSeries), click the **WebSphere MQ for Windows** link. The WebSphere MQ Support page opens.

   This page displays only the most recent fix pack for WebSphere MQ. However, previous fix packs can still be downloaded.
4. Click the link to download the most recent fix pack for WebSphere MQ for Windows V5.3. The Fix Pack and Interim Fixes page opens.
5. Select **CSD04**, then continue through the download and registration process to download the setup files.
6. When the setup files have been downloaded, read the accompanying documentation and release notes, then start the installer and install the CSD.

### *Installing a Java Runtime Environment*

Although the WebSphere Business Integration Message Brokers preinstall launchpad does not check for a Java Runtime Environment (JRE), JRE 1.3.1 or later must be installed to run the IBM Agent Controller installer. The IBM 32-bit SDK for Java 2 V1.4.0 is supplied on the WebSphere MQ V5.3 CD.

To install a JRE:

1. Insert the CD that is labeled *IBM WebSphere MQ for Windows Version 5.3* and click **OK**.

   If the CD does not run automatically, use Windows Explorer to browse to the CD, and double-click **MQLaunch.exe**. The WebSphere MQ Installation Launchpad opens.

2. On the Software Prerequisites page of the launchpad, click the plus sign (**+**) next to **Supported Java Runtime Environment 1.3 or later** to expand that section of the page.
3. Click **WebSphere MQ CD**. The installation starts.
4. Work through the installer to complete the installation of the JRE.

### *Installing IBM DB2 UDB Version 8.1 with FixPak 2*

Although WebSphere Business Integration Message Brokers can be used with IBM DB2 UDB Version 7.2 with FixPak 9, IBM DB2 UDB Version 8.1 is supplied in the WebSphere Business Integration Message Brokers product package. However FixPak 2 must have been downloaded from the IBM Web site and installed separately.

To install DB2 V8.1:

1. In the launchpad, click **CD-ROM**.
2. When prompted, insert the CD that is labeled *DB2 V8.1 for WebSphere Business Integration broker products (Windows NT, Windows 2000, and Windows XP)*, then click **OK**. The CD autoruns.

   If the CD does not run automatically, use Windows Explorer to browse to the CD, and double-click **Setup.exe**.

   The IBM DB2 Setup Launchpad opens.
3. In the launchpad, verify that the correct installation prerequisites to install DB2 are there, and read the Release Notes.
4. Click **Install Products** to start the installation.
5. When prompted, select the following options:
   – **Typical** installation.
   – Accept all of the default values. This gives an installation that is sufficient for use with WebSphere Business Integration Message Brokers.
   – Ensure that the user account that is specified to enable DB2 administration server to access the system has Administrator privileges.
   – On the “Specify a contact for health monitor notification” page, select **Defer the task until after installation is complete**.

For further instructions, refer to the DB2 product documentation and readme.

To install the DB2 V8.1 FixPak 2:

1. Stop all DB2 services using the following commands:

```
db2stop
db2admin stop
```

2. In a Web browser, go to the following URL:

   `http://www-306.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report`

   This is the IBM Web site from which DB2 FixPaks can be downloaded.
3. Select the latest Windows 32-bit FixPak.
4. Follow the instructions on the Web page to download and install the FixPak (DB2 Enterprise Server edition). The downloaded file is about 600 MB.
5. Read the Release notes and installation readme document.
6. When the setup files have been downloaded, start the installer and install the FixPak.

### *Installing IBM Agent Controller*

Installation of IBM Agent Controller is necessary if the flow debugger in WebSphere Business Integration Message Brokers is required to debug message flow applications. The IBM Agent Controller setup files are supplied in the product package. Before installing IBM Agent Controller, a Java Runtime Environment (JRE) 1.3.1 or later must be installed.

To install IBM Agent Controller:

1. In the launchpad, click **CD-ROM**.
2. When prompted, insert the CD that is labeled *WebSphere Business Integration Event Broker Supplemental* or *WebSphere Business Integration Message Broker Supplemental*, then click **OK**.

   If the CD does not run automatically, use Windows Explorer to browse the CD to the *Windows\RAC* folder, then double-click **Setup.exe** to start the installation.
3. When prompted, select the following options:
   - When specifying the location of the JRE, browse to the /bin directory, which contains the java.exe file.
   - **Disable** security.

Table 3-3 on page 56 lists the contents of each CD that is supplied in the product package for installing WebSphere Business Integration Event Broker.

## Confirming that the number of license units is correct

To use WebSphere Business Integration Message Brokers, the correct number of license units for the number of processors on your computer must be available.

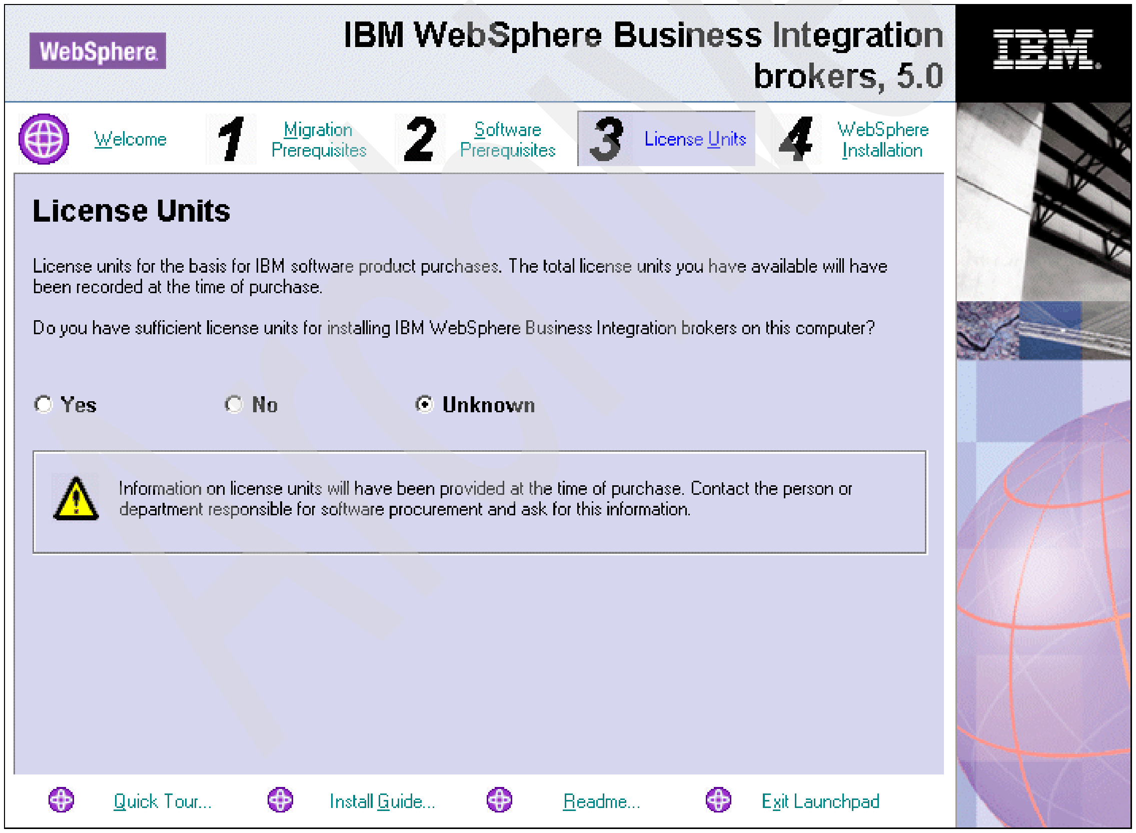
To confirm that the correct number of license units is available:

1. Click **License units**, which is also labeled **3**. The license units page is displayed, as shown in Figure 3-5.

   When purchasing WebSphere Business Integration Message Brokers, the licenses that are required to use the product also must have been purchased. Contact the person or department responsible for software procurement to obtain this license information.

2. Confirm whether the number of license units is correct:

   If you click **No** to confirm that sufficient licenses to use the product have not been purchased or **Unknown** because license information cannot be ascertained, contact the person or department responsible for software procurement to organize purchase of licenses. WebSphere Business Integration Message Brokers can still be installed, but the correct number of license units must be obtained before the product can be used.



License Units

License units for the basis for IBM software product purchases. The total license units you have available will have been recorded at the time of purchase.

Do you have sufficient license units for installing IBM WebSphere Business Integration brokers on this computer?

Yes No [✓ Unknown]

Information on license units will have been provided at the time of purchase. Contact the person or department responsible for software procurement and ask for this information.

*Figure 3-5 License Units page*

## 3.2 Installing Message Broker or Event Broker

When the required prerequisite software has been installed, the installation of WebSphere Business Integration Event Broker or WebSphere Business Integration Message Broker can commence.

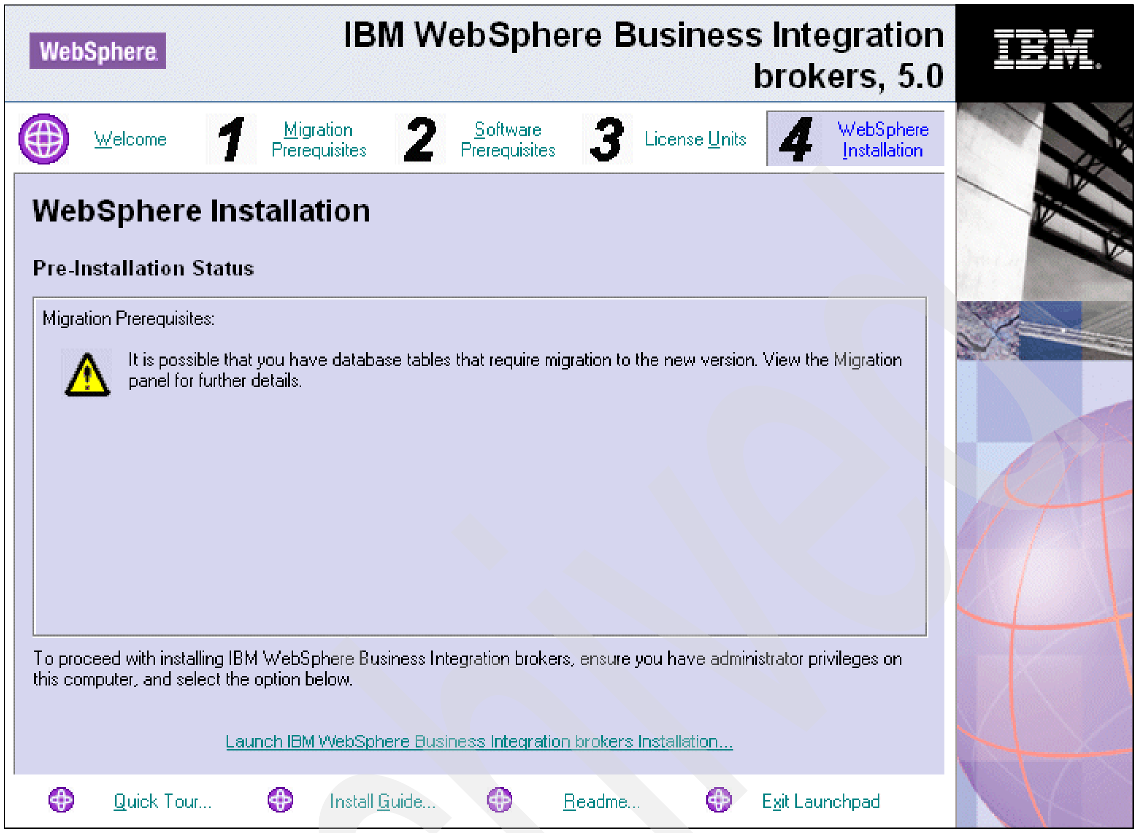
### 3.2.1 Starting the installer

If you are not using the preinstall launchpad as a guide through the installation process, start the installer directly from the CD:

1. Insert the CD that is labeled *WebSphere Business Integration Message Broker for Windows NT, Windows 2000, and Windows XP version 5.0.2*.
2. Browse the CD and double-click the file called **Setup.exe** to start the installer.

If you have been using the preinstall launchpad, start the installer from the launchpad:

1. In the launchpad, click **WebSphere Installation**, which is also labeled **4**. The WebSphere Installation page is displayed, as shown in Figure 3-6 on page 38.



WebSphere Installation

Pre-Installation Status

Migration Prerequisites:

It is possible that you have database tables that require migration to the new version. View the Migration panel for further details.

To proceed with installing IBM WebSphere Business Integration brokers, ensure you have administrator privileges on this computer, and select the option below.

Launch IBM WebSphere Business Integration brokers Installation...

*Figure 3-6 WebSphere Installation page*

2. Read and respond to the messages that are displayed on the WebSphere Installation page, which might include the following issues:

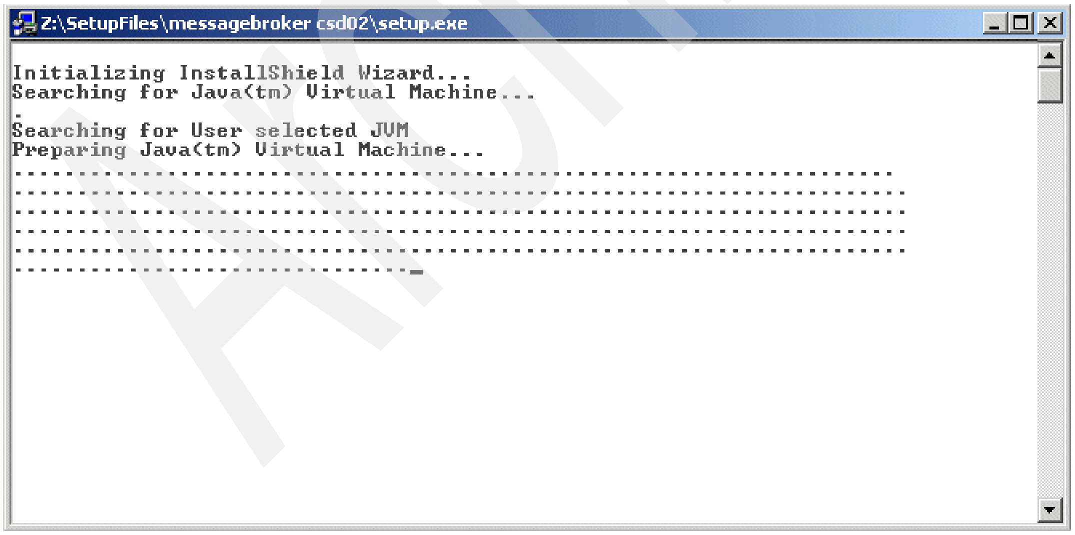
   – The launchpad cannot detect whether database tables relating to WebSphere Business Integration Message Brokers exist on the computer. The launchpad displays a warning message and asks whether the database tables on the computer from previous versions of WebSphere Business Integration Message Brokers (if there are any) are to be used with the new installation. Migration steps that are referred to in the most recent version of the Install Guide must be performed.

   – The launchpad also warns if you have not yet installed some of the prerequisite software. You can continue with the installation of the product without satisfying all prerequisites; in fact, you might want to if you are installing only the Message Brokers Toolkit. You can install remaining prerequisite software later. However, be aware that the product cannot function fully if some prerequisite products are missing. For details about which prerequisite software to install for specific functions of WebSphere Business Integration Message Brokers, see Table 3-1 on page 24.

3. Ensure that you are logged on to the computer as a user with administrative privileges but *not* as the user Administrator.
4. If you are installing from a network location, ensure that the directory containing the setup files is mapped to your computer. Otherwise, the installer fails and very briefly displays an error message before exiting.
5. Click the **Launch IBM WebSphere Business Integration Brokers Installation** link at the bottom of the WebSphere Installation page.
6. When prompted, insert the appropriate CD:
   - If you are installing WebSphere Business Integration Event Broker, insert the CD labeled *WebSphere Business Integration Event Broker for Windows NT, and Windows XP version 5.0.2*.
   - If you are installing WebSphere Business Integration Message Broker, insert the labeled *WebSphere Business Integration Message Broker for Windows NT, and Windows XP version 5.0.2*.

## 3.2.2 Installing the product

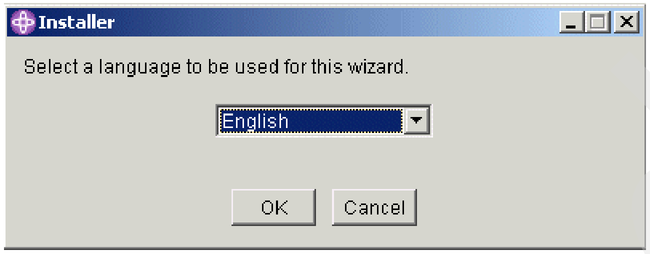
When the installer starts, a command window opens while the Java Virtual Machine initializes (Figure 3-7). Do not perform other tasks on the machine while this is taking place.



```
Initializing InstallShield Wizard...
Searching for Java(tm) Virtual Machine...
.
Searching for User selected JVM
Preparing Java(tm) Virtual Machine...
.......................................................................
........................................................................
........................................................................
........................................................................
........................................................................
................................_
```

*Figure 3-7 Initializing the Java Virtual Machine*

1. In the Installer window, select the language in which you want the installer to be displayed, and click **OK** (Figure 3-8). If this dialog is not displayed, it might be hidden behind another window on the desktop.

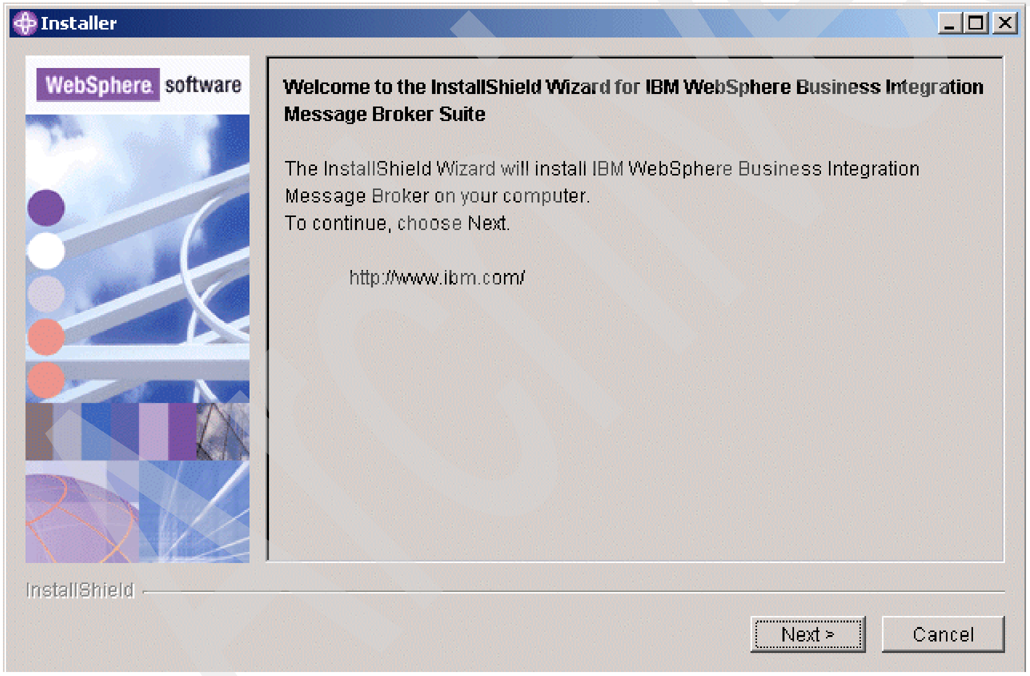

Installer

Select a language to be used for this wizard.

English

*Figure 3-8 Installer language selection dialog*

2. The first page of the installer is displayed (Figure 3-9). Click **Next** to continue.


Installer

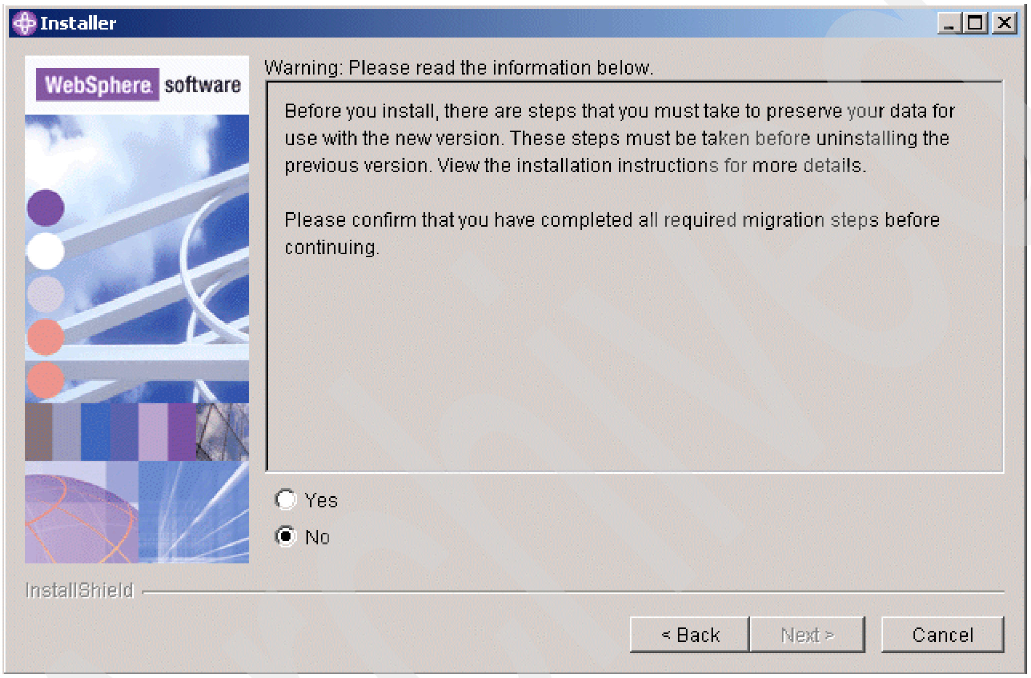
**Welcome to the InstallShield Wizard for IBM WebSphere Business Integration Message Broker Suite**

The InstallShield Wizard will install IBM WebSphere Business Integration Message Broker on your computer.
To continue, choose Next.

http://www.ibm.com/

*Figure 3-9 Installer Welcome page*

3. Confirm that you have completed all of the required migration tasks (Figure 3-10):
   – To continue with the installation, click **Yes**, then click **Next**.
   – To cancel the installation and perform the required migration tasks, click **No**, then click **Cancel**. If you are migrating from a previous version of the product refer to the *most recent version* of the Install Guide. (To obtain it, open the launchpad, then click the **Install Guide** button).



Warning: Please read the information below.

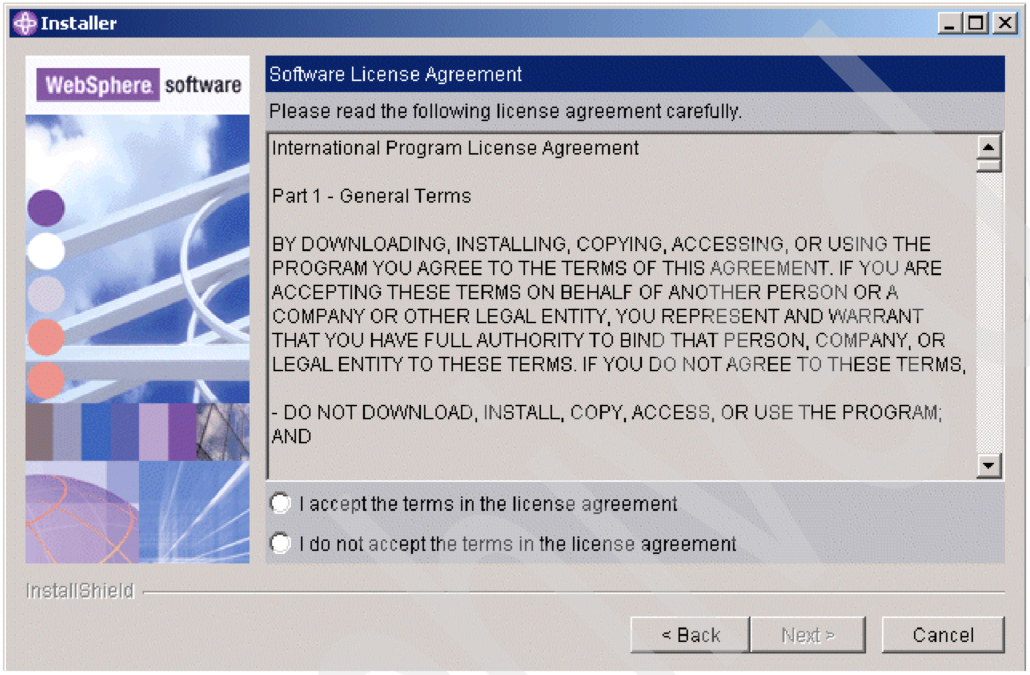
Before you install, there are steps that you must take to preserve your data for use with the new version. These steps must be taken before uninstalling the previous version. View the installation instructions for more details.

Please confirm that you have completed all required migration steps before continuing.

Yes [✓ No]

*Figure 3-10 Migration prerequisites page*

4. Read the Software License Agreement (Figure 3-11) and click one of the options to signal your acceptance or rejection of the agreement. You cannot continue with the installation if you do not accept the Software License Agreement.



Please read the following license agreement carefully.

International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

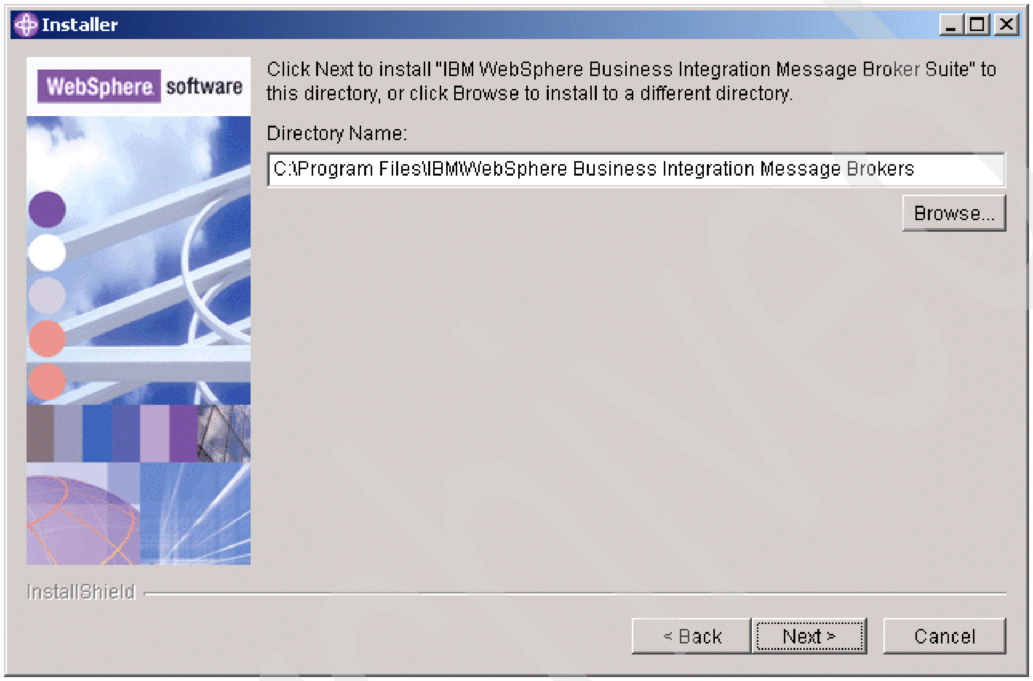
- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

○ I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

*Figure 3-11 Software License Agreement page*

5. Decide where to install WebSphere Business Integration Message Brokers (Figure 3-12):
   – To install the product in the default location, click **Next**.
   – To browse to an alternate location, click **Browse**. When you have selected the location, click **Next**.



Click Next to install "IBM WebSphere Business Integration Message Broker Suite" to this directory, or click Browse to install to a different directory.

Directory Name:

C:\Program Files\IBM\WebSphere Business Integration Message Brokers

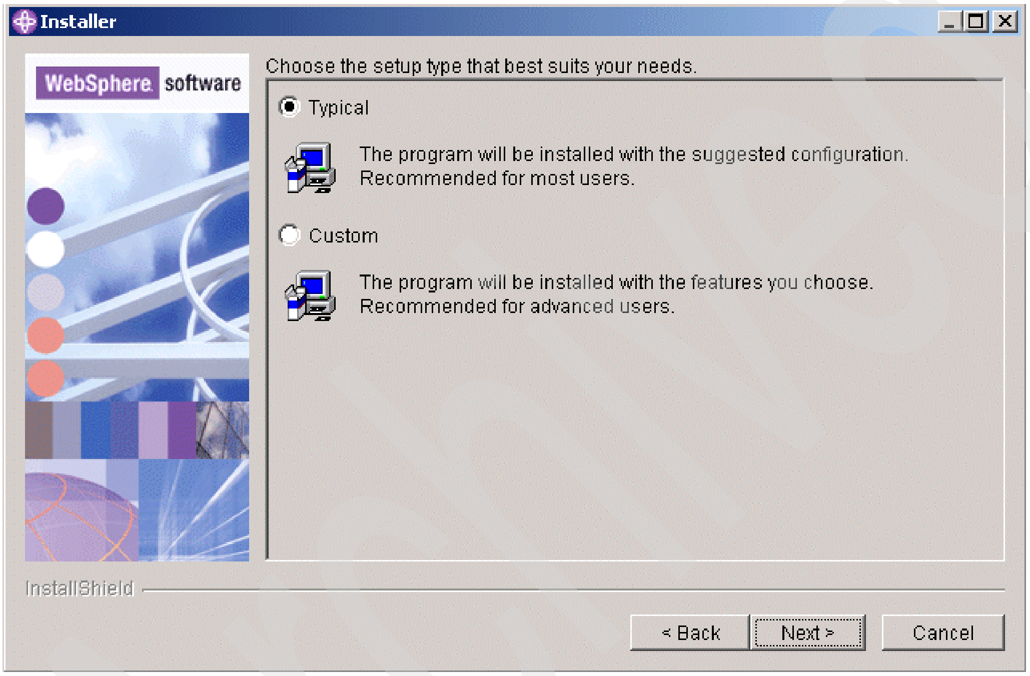
*Figure 3-12   The Install location page*

6. Select the type of installation that you want to perform (Figure 3-13):

   – To install all of the components that you need, click **Typical**. This option is recommended for most users.

   – To select specific components, click **Custom**. Select this option only when you are more familiar with the product components.

   See Chapter 1, “Introduction and overview” on page 1 for an introduction and overview of the main components of Event Broker and Message Broker.



Choose the setup type that best suits your needs.

[✓ Typical]

The program will be installed with the suggested configuration.
Recommended for most users.

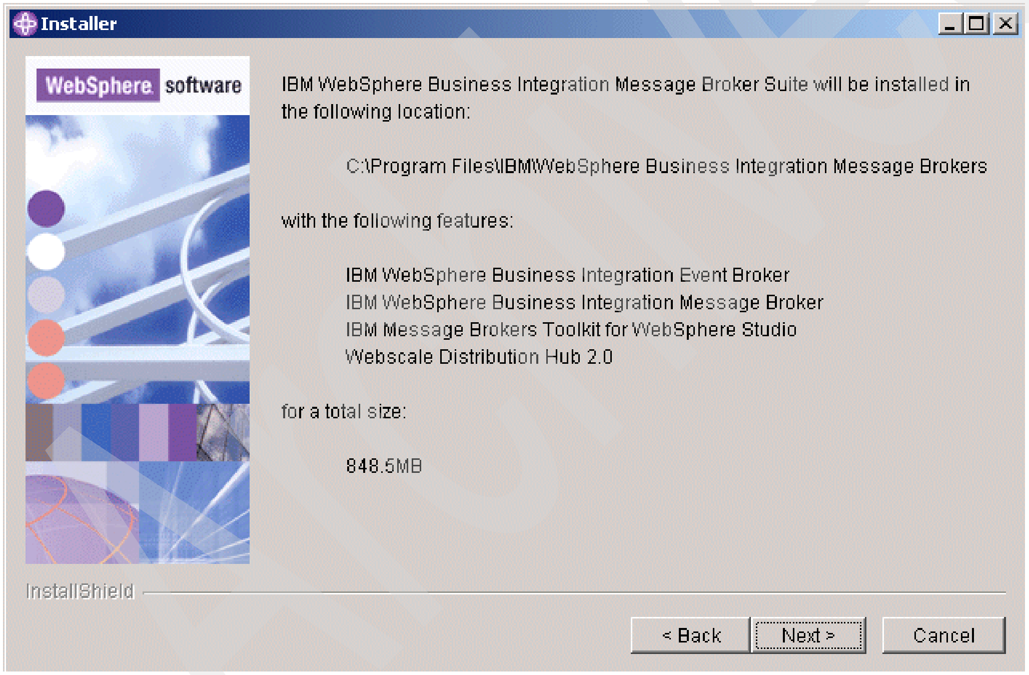
Custom

The program will be installed with the features you choose.
Recommended for advanced users.

*Figure 3-13 Install type selection page*

The installer displays a summary screen of the products that are to be installed (Figure 3-14).

– If you are installing Event Broker, the following components are installed:
  - WebSphere Business Integration Event Broker
  - IBM Message Brokers Toolkit for WebSphere Studio
  - Webscale Distribution Hub 2.0
– If you are installing Message Broker, these components are installed:
  - WebSphere Business Integration Event Broker
  - WebSphere Business Integration Message Broker
  - IBMMessage Brokers Toolkit for WebSphere Studio
  - Webscale Distribution Hub 2.0



IBM WebSphere Business Integration Message Broker Suite will be installed in the following location:

C:\Program Files\IBM\WebSphere Business Integration Message Brokers

with the following features:

IBM WebSphere Business Integration Event Broker
IBM WebSphere Business Integration Message Broker
IBM Message Brokers Toolkit for WebSphere Studio
Webscale Distribution Hub 2.0

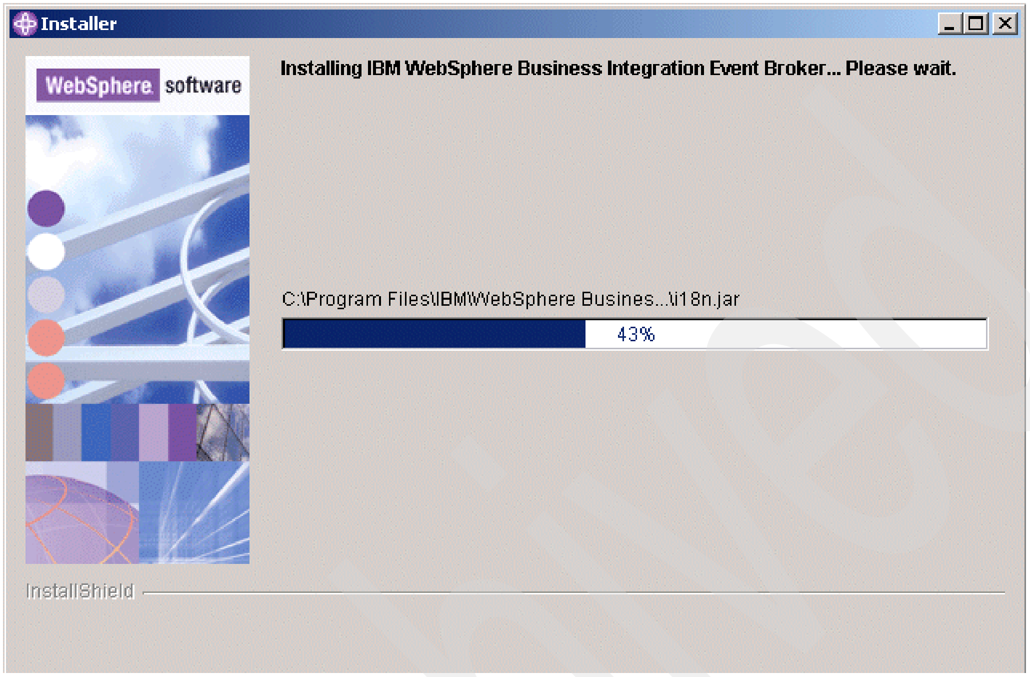
for a total size:

848.5MB

*Figure 3-14 Installation summary page*

7. Click **Next** to install the displayed components.

A progress bar is displayed (Figure 3-15, which shows the Event Broker installer).



Installing IBM WebSphere Business Integration Event Broker... Please wait.

C:\Program Files\IBM\WebSphere Busines...\i18n.jar

43%

*Figure 3-15 WebSphere Business Integration Event Broker installation progress*

8. A separate progress page is displayed for each part of the software:
   - If you are installing Event Broker, these progress screens are displayed:
     - WebSphere Business Integration Event Broker
     - WebSphere Business Integration Event Broker uninstaller
     - IBM Message Brokers Toolkit for WebSphere Studio
     - IBM Message Brokers Toolkit for WebSphere Studio uninstaller
   - When installing Message Broker, these progress screens are displayed:
     - WebSphere Business Integration Event Broker
     - WebSphere Business Integration Event Broker uninstaller
     - WebSphere Business Integration Message Broker
     - WebSphere Business Integration Message Broker uninstaller
     - IBM Message Brokers Toolkit for WebSphere Studio.
     - IBM Message Brokers Toolkit for WebSphere Studio uninstaller.

## Configuring security groups

When the installer finishes installing WebSphere Business Integration Message Brokers, the Security wizard is launched automatically. The wizard creates the user groups that are needed to control the security of WebSphere Business Integration Message Brokers resources and tasks. The wizard also assigns an administrator's user account to the relevant groups.

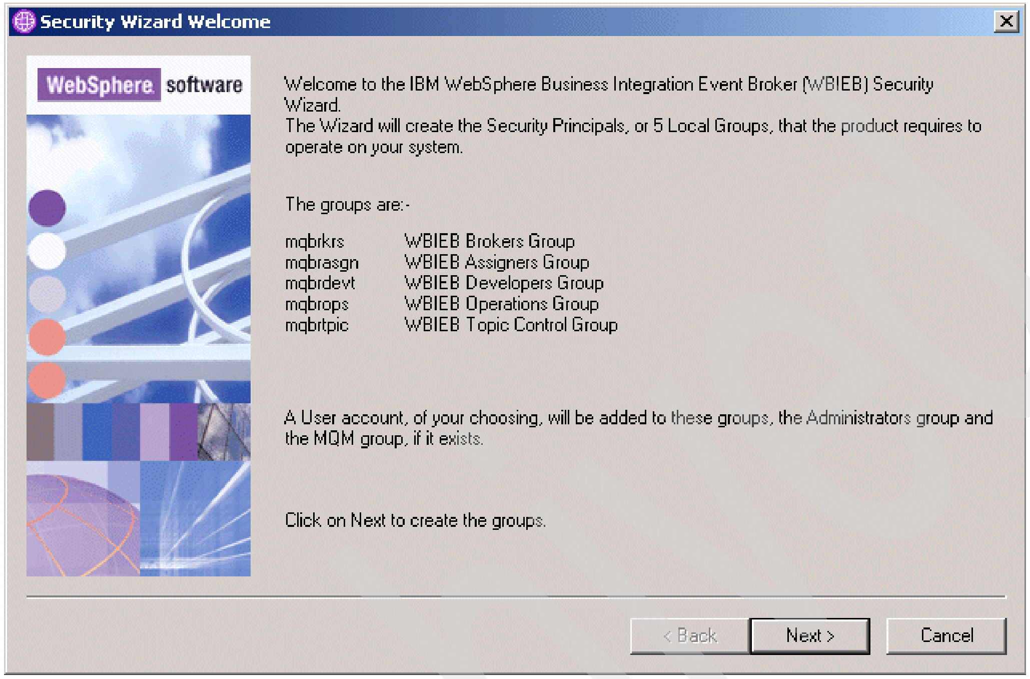
The Security wizard creates the security groups shown in Table 3-2.

*Table 3-2 The security groups that are created by the Security Wizard*

| Group | Purpose |
|---|---|
| mqbrasgn | Enables users to assign message flows and message sets to brokers. |
| mqbrdevt | Enables users to create message sets and message flows. |
| mqbrkrs | Enables users to start brokers, the User Name Server, and the Configuration Manager. |
| mqbrops | Enables users to manage brokers, message flows, and broker domains; view logs; trace message flows. |
| mqbrtpic | Enables users to manage topics and access control lists. |

To create and assign a user account to the relevant security groups:

1. On the Welcome page of the Security wizard, click **Next** (Figure 3-16).



Welcome to the IBM WebSphere Business Integration Event Broker (WBIEB) Security Wizard.
The Wizard will create the Security Principals, or 5 Local Groups, that the product requires to operate on your system.

The groups are:-

| | |
|---|---|
| mqbrkrs | WBIEB Brokers Group |
| mqbrasgn | WBIEB Assigners Group |
| mqbrdevt | WBIEB Developers Group |
| mqbrops | WBIEB Operations Group |
| mqbrtpic | WBIEB Topic Control Group |

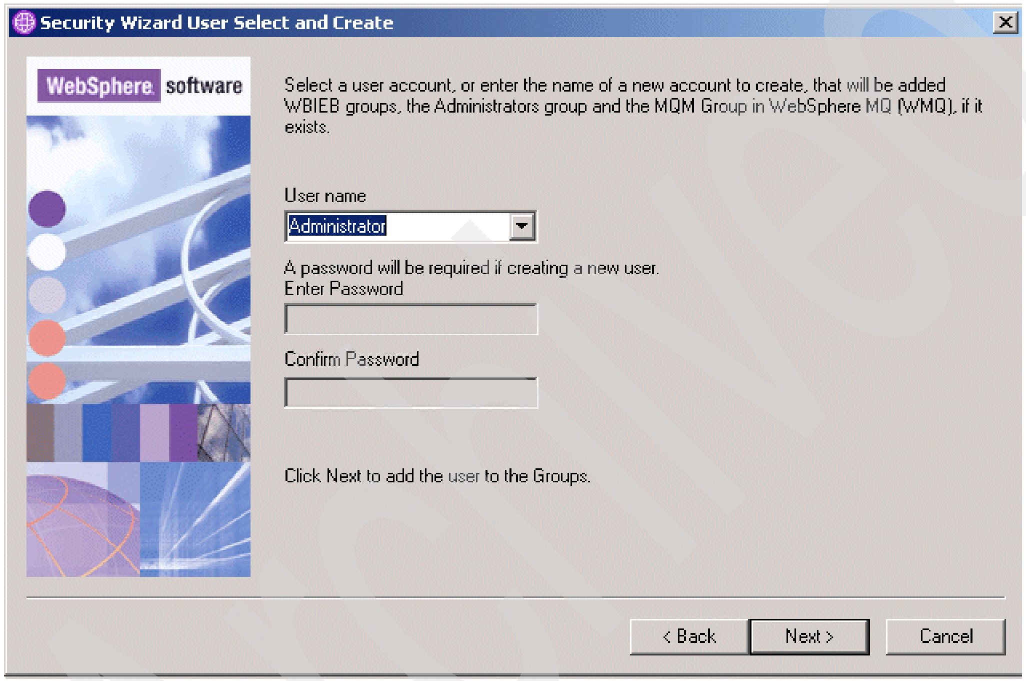
A User account, of your choosing, will be added to these groups, the Administrators group and the MQM group, if it exists.

Click on Next to create the groups.

*Figure 3-16 Security Wizard Welcome*

2. From the User name list, select the user account to add to all of the relevant WebSphere Business Integration Message Brokers security groups, then click **Next** (Figure 3-17).

   **Important:** Do not select the user name called Administrator. You cannot run WebSphere Business Integration Message Brokers using the default Windows Administrator account.



Select a user account, or enter the name of a new account to create, that will be added WBIEB groups, the Administrators group and the MQM Group in WebSphere MQ (WMQ), if it exists.

User name: Administrator

A password will be required if creating a new user.

Enter Password

Confirm Password

Click Next to add the user to the Groups.

*Figure 3-17 The User Select and Create page*

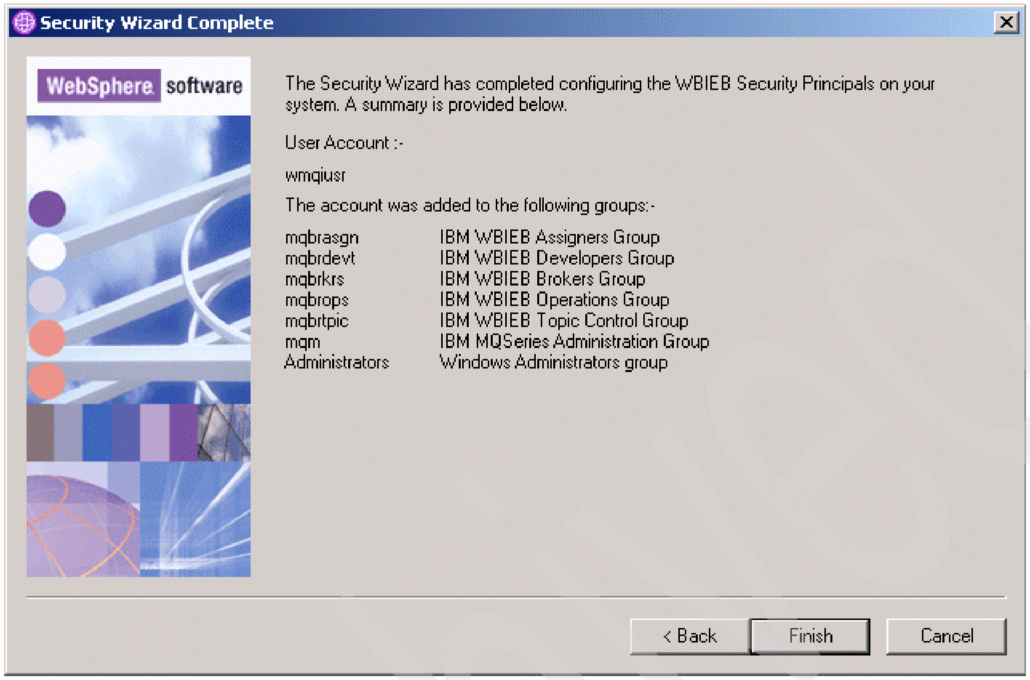
   The user account is added to all of the groups listed in Table 3-2 on page 47. The user account is also added to the following groups:

   – mqm, the WebSphere MQ security group.

   – Administrators, the Windows administrators group.

3. Optional: To add another user to the groups, click **Back** to add another user to the WebSphere Business Integration Message Brokers security groups.

4. On the summary page (Figure 3-18), click **Finish**.



The Security Wizard has completed configuring the WBIEB Security Principals on your system. A summary is provided below.

User Account :-

wmqiusr

The account was added to the following groups:-

| | |
|---|---|
| mqbrasgn | IBM WBIEB Assigners Group |
| mqbrdevt | IBM WBIEB Developers Group |
| mqbrkrs | IBM WBIEB Brokers Group |
| mqbrops | IBM WBIEB Operations Group |
| mqbrtpic | IBM WBIEB Topic Control Group |
| mqm | IBM MQSeries Administration Group |
| Administrators | Windows Administrators group |

*Figure 3-18 Security Wizard summary page*

## 3.3 Post-installation tasks

Before you can use the product, and before you create and start a broker, complete the following tasks:

- Confirm that you have enough licenses for the number of processors on the computer. You cannot start the broker if you do not have enough licenses.
- Register the product. This is required by the license.

### 3.3.1 Confirming that you have enough licenses

You must have enough licenses for the number of processors on the computer before you can start a broker.

To confirm that you have enough licenses:

1. At a command prompt, enter the following command:

```
mqsidisplaycapacity
```

The number of processors installed on the computer is displayed.

2. If you have enough licenses for the number of processors installed on the computer, enter the following command:

   ```
   mqsisetcapacity -c n
   ```

   - *n* is the number of processors installed on the computer.

### 3.3.2 Registering the product

To fulfill licensing requirements, you must register your copy of WebSphere Business Integration Message Brokers with IBM Software Delivery and Fulfillment.

To register the product installation, run the product registration tool. The product registration tool is supplied in the following locations.

#### Registering Event Broker

To register WebSphere Business Integration Event Broker Version 5.0.2:

1. In the installation directory, double-click **prt\event\PRT5724E11.exe**. The product registration form is displayed.
2. Complete the following information:
   – Contact information
   – System information
   – Additional information
3. Click **Submit** to send the information to IBM.

   If you do not have an Internet connection, select one of the following options:

   – The product registration tool runs again the next time the machine is rebooted.
   – Print the form and register by mail or by fax. A print screen opens. Select a suitable print format and printer (the printed document includes fax and mail address details).

If you click Exit at any time during the process, you will be prompted to register in future.

The product registration tool is also supplied on the WebSphere Business Integration Event Broker Version 5.0.2 CD.

### Registering Message Broker

To register WebSphere Business Integration Message Broker Version 5.0.2:

1. In the installation directory, double-click **prt\event\PRT5724E26.exe**. The product registration form is displayed.
2. Complete the following information:
   - Contact information
   - System information
   - Additional information
3. Click **Submit** to send the information to IBM.

   If you do not have an Internet connection, select one of the following options:
   - The product registration tool runs again the next time the machine is rebooted.
   - Print the form and register by mail or by fax. A print screen opens. Select a suitable print format and printer (the printed document includes fax and mail address details).

If you click Exit at any time during the process, you will prompted to register in future.

The product registration tool is also supplied on the WebSphere Business Integration Message Broker Version 5.0.2 CD.

## 3.4 Installing product fix packs

The instructions in this section describe how to install a fix pack on top of an installed version of WebSphere Business Integration Message Brokers. You might have obtained a fix pack on CD, or you can download the files from the Web. A link to the service download site can be found in the Useful links section of Appendix A, "Getting help" on page 219.

### 3.4.1 Before you install the fix pack

Before installing the fix pack, perform the following tasks:

1. Ensure that the user ID with which you are logged on is a member of the Windows Administrators group in the local security domain. Use the same user ID that was used to install WebSphere Business Integration Message Brokers.

2. Ensure that all WebSphere Business Integration Message Brokers services are stopped, including the Configuration Manager, the User Name Server (if applicable), and all brokers on the system:
   - To stop the Configuration Manager, enter the following command at a command prompt:
     ```
     mqsistop configmgr
     ```
   - To stop each broker, enter the following command (replacing *WBRK_BROKER* with the name of the broker):
     ```
     mqsistop WBRK_BROKER
     ```
   - To stop the User Name Server, enter the following command:
     ```
     mqsistop usernameserver
     ```
3. Close the Message Brokers Toolkit.
4. Close all WebSphere Business Integration Message Brokers files, such as the product Readme.
5. Close all running programs.

## 3.4.2 Installing the fix pack

The following instructions describe how to install the fix pack. You can install the fix pack from a CD or from files downloaded from the Web.

### Installing the fix pack from a CD

To install the fix pack from a CD:

1. Read the contents of the memo.ptf file and any readme.html files. These files are in the root directory of the CD and contain additional information regarding the installation and fixes available in this fix pack.
2. Insert the fix pack CD into the CD-ROM drive. If autorun is enabled, setup.exe runs automatically. If not, browse the CD and double-click **setup.exe**.

### Installing the fix pack from files downloaded from the Web

To install the fix pack from files that are downloaded from the Web:

1. Read the contents of the memo.ptf file that accompanies the downloaded file for additional information about installation and fixes available in this fix pack.
2. Run the executable file that was downloaded.
3. When prompted, accept the default install location or specify an alternative.
4. Click **Next** to extract the files.
5. Click **Finish** to close the dialog. The setup.exe file runs automatically.

## 3.4.3 Known issue with installing a fix pack on WebSphere Business Integration Message Brokers version 5.0.0

There is a known issue that can result in errors in the Message Brokers Toolkit. The errors occur if:

1. You have installed WebSphere Business Integration Message Brokers version 5.0.0 (the General Availability version).
2. You have used the Message Brokers Toolkit.
3. You have subsequently installed any WebSphere Business Integration Message Brokers fix pack.

When you open the Message Brokers Toolkit, the Update Manager is launched correctly but any updates to your plug-ins fail. Consequently, when the Message Brokers Toolkit opens after the updates have been installed, it opens with the incorrect plug-in levels. You then get errors because of plug-in incompatibilities.

To verify that the levels of plug-ins that you are using are correct:

1. In the Message Brokers Toolkit, click **Help** → **About Message Brokers Toolkit for WebSphere Studio**.
2. Under the heading Message Brokers Toolkit for WebSphere Studio, a version number is displayed.

If the version number that is displayed is 5.0.*x*. where *x* is the number of the fix pack installed, the Message Brokers Toolkit is using the correct plug-in versions after you have installed a fix pack.

If the version that is displayed is less than 5.0.*x*, do the following:

1. Close the Message Brokers Toolkit.
2. In the installation directory, navigate to eclipse\workspace and delete the .metadata directory.
3. Start the Message Brokers Toolkit.

If you have existing WebSphere Business Integration Message Brokers resources, such as message flows and message sets, you must re-import them into the workspace.

To import existing projects into the workspace:

1. In the Message Brokers Toolkit, click **File** → **Import**.
2. In the Import wizard, click **Existing Project into Workspace**, then click **Next**.
3. Click **Browse** to locate the project in the workspace directory of WebSphere Business Integration Message Brokers.

4. Click **Finish**.
5. Repeat steps 1 on page 54 through 4 to import each project.

With the fix pack installed, you can run the Message Brokers Toolkit, configure the environment, and develop applications.

## 3.5 Creating a default configuration

With WebSphere Business Integration Event Broker or WebSphere Business Integration Message Broker installed, you can configure the product and develop applications.

Before you can run the Getting Started samples to verify that the installation of WebSphere Business Integration Message Brokers has been successful, you must configure the product.

You can start to develop message sets and flows without first creating any runtime components, but you cannot test or run your applications or the provided samples. For more information about developing message flows and message sets, see Chapter 4, “Application development” on page 83.

To run the samples, message flows, and any associated message sets, you must create a broker domain.

A broker domain represents the WebSphere Business Integration Message Broker or WebSphere Business Integration Event Broker runtime environment. A broker domain contains a single Configuration Manager with associated brokers. It may also contain a User Name Server if authentication is used for publish/subscribe messaging.

To quickly create a default configuration for WebSphere Business Integration Message Brokers, run the Getting Started wizard. When you are more familiar with WebSphere Business Integration Message Brokers, you can create broker domains manually.

To learn more about broker domains and how to create the components of a broker domain manually, see Chapter 5, “Administration” on page 133. In particular, see 5.1.1, “Manually creating a basic Message Broker configuration” on page 134, which explains how to manually create the components and resources for a basic broker domain configuration.

## What does the Getting Started wizard do?

The Getting Started wizard creates the minimum components to run a simple message flow application. To run the Getting Started wizard, you must have installed the prerequisite software such as DB2 and WebSphere MQ.

The wizard creates:

- A broker
- A DB2 database for storing the Configuration Manager's data
- A DB2 database for storing data that is deployed to the broker
- A link between the domain and the broker
- A WebSphere MQ queue manager, which is used by the broker and the Configuration Manager
- A configuration for the queue manager to use a listener port for communication between the Configuration Manager and the Message Brokers Toolkit
- A project within the Message Brokers Toolkit workspace to contain the broker domain connection information

Table 3-5 lists the components that the wizard creates and their default values.

*Table 3-3 Components created by the Getting Started wizard*

| **Component** | **Default value** |
|---|---|
| WebSphere Business Integration Brokers services user account | Your Windows account |
| WebSphere Business Integration Brokers services DB2 account | Your Windows account |
| Broker domain and broker domain connection | Localdomain |
| Server Project | LocalProject |
| Queue Manager name | WBRK_QM |
| Queue Manager port | By default the field for this information is empty. The port number conventionally has four digits, that is 1414. |
| Configuration Manager | Configuration Manager<br>You cannot change the name of the Configuration Manager. |
| Configuration manager database | `WBRKCMDB` |

| Component | Default value |
|---|---|
| Broker | WBRK_BROKER |
| Broker database | WBRKBKDB |

## Starting the Message Brokers Toolkit

Before you can run the Getting Started wizard, open the Message Brokers Toolkit by clicking **Start** → **Programs** → **IBM WebSphere Studio** → **Message Brokers Toolkit**.

Wait while the installation completes (Figure 3-19).

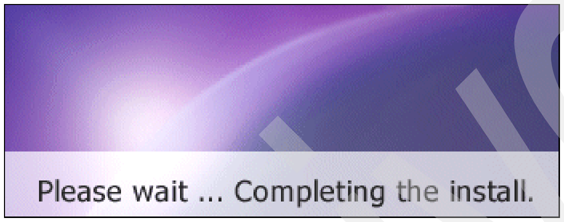
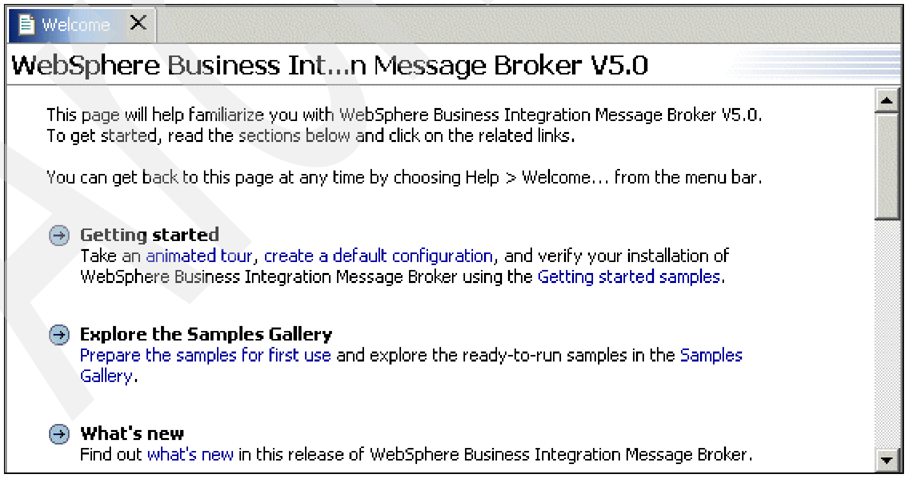

Please wait ... Completing the install.

*Figure 3-19 Install completion screen*

When installation completes, the Message Brokers Toolkit opens.

The welcome information provided on the opening screen of the Message Brokers Toolkit provides Getting Started information (Figure 3-20).



WebSphere Business Int...n Message Broker V5.0

This page will help familiarize you with WebSphere Business Integration Message Broker V5.0.
To get started, read the sections below and click on the related links.

You can get back to this page at any time by choosing Help > Welcome... from the menu bar.

**Getting started**
Take an animated tour, create a default configuration, and verify your installation of WebSphere Business Integration Message Broker using the Getting started samples.

**Explore the Samples Gallery**
Prepare the samples for first use and explore the ready-to-run samples in the Samples Gallery.

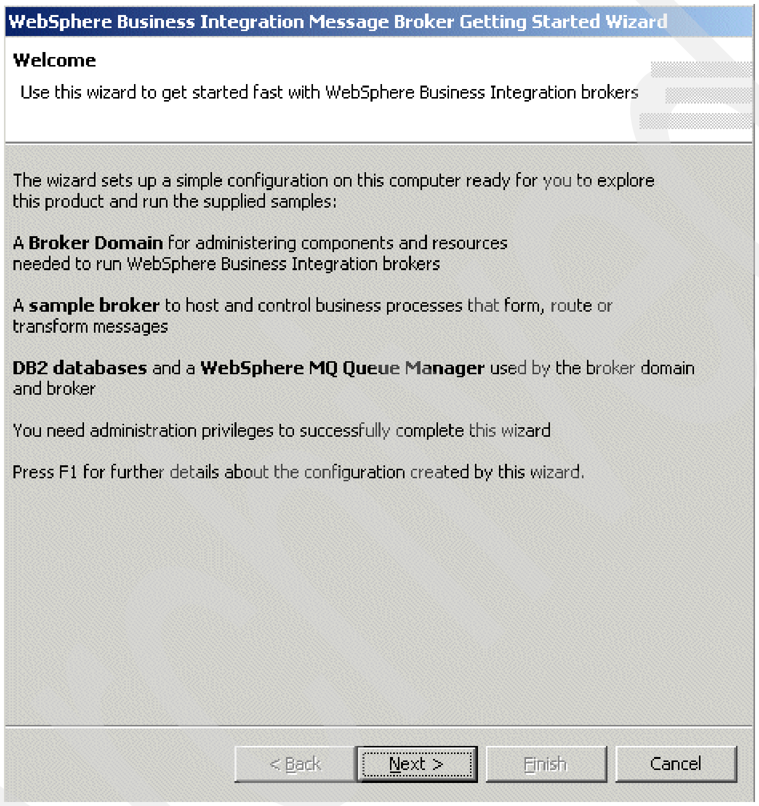
**What's new**
Find out what's new in this release of WebSphere Business Integration Message Broker.

*Figure 3-20 Message Broker tooling welcome page*

## Running the Getting Started wizard

To create a default configuration using the Getting Started wizard:

1. Ensure that you are logged on with administrator privileges.
2. On the product welcome page, click **Create a default configuration**. The Getting Started wizard opens (Figure 3-21).


WebSphere Business Integration Message Broker Getting Started Wizard
< Back
Next >
Finish
Cancel

**Welcome**

Use this wizard to get started fast with WebSphere Business Integration brokers

The wizard sets up a simple configuration on this computer ready for you to explore this product and run the supplied samples:

A **Broker Domain** for administering components and resources needed to run WebSphere Business Integration brokers

A **sample broker** to host and control business processes that form, route or transform messages

**DB2 databases** and a **WebSphere MQ Queue Manager** used by the broker domain and broker

You need administration privileges to successfully complete this wizard

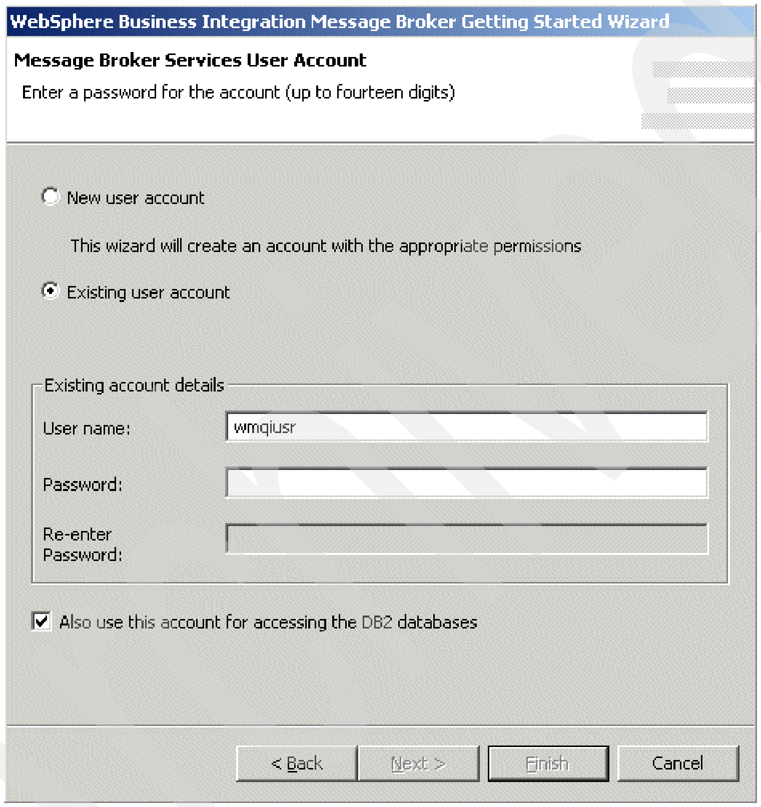
Press F1 for further details about the configuration created by this wizard.

*Figure 3-21 Getting Started wizard welcome page*

3. Click **Next**.
4. Enter the information for the user account to use to run the WebSphere Business Integration Message Brokers services, such as the broker (Figure 3-22 on page 59):
   – To create a new account, select **New user account**, then follow the steps to create the new account.

– To use an existing account, select **Existing user account**, then enter the user name and password for the account that you want to use.

  The user account must be a member of the Administrators group and the mqm and mqbrkrs security groups. The account must also have enabled the security privilege to "act as part of the operating system." If the account does not have this privilege, the Getting Started wizard can set it but it requires you to reboot the computer.


WebSphere Business Integration Message Broker Getting Started Wizard

Message Broker Services User Account

Enter a password for the account (up to fourteen digits)

( ) New user account

This wizard will create an account with the appropriate permissions

(•) Existing user account

Existing account details

User name: wmqiusr

Password:

Re-enter Password:

☑ Also use this account for accessing the DB2 databases

*Figure 3-22 Message Broker Services User Account page*

5. Decide which user account to use to access the broker and Configuration Manager DB2 databases:

   – To use the same user account as in step 4 on page 58, ensure that the **Also use this account for accessing the DB2 databases** check box is selected.

   – To use a different account, clear the **Also use this account for accessing the DB2 databases** check box. Click **Next** and complete the information on the next page to specify the account to be used for accessing the DB2 databases.

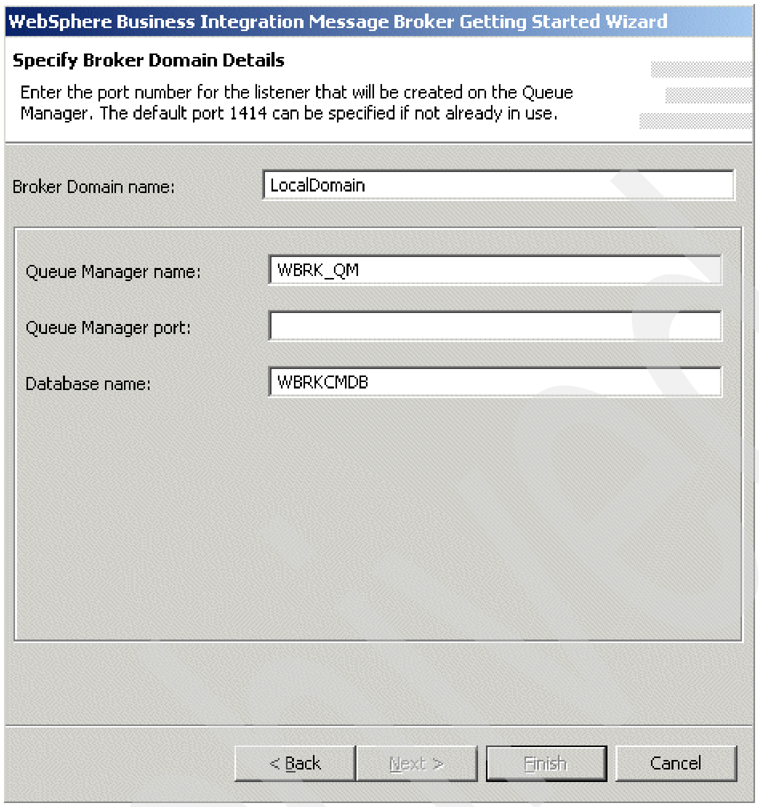
6. Click **Next**.
7. In the Queue Manager port field (Figure 3-23 on page 61), type the number of the port on which the queue manager listens. Type `1414` unless another queue manager is already using that port. Port 1414 is the default WebSphere MQ port.

   To verify which ports are already in use on the processor, open a command prompt and enter the following command:

   ```
   netstat -a
   ```

   If port 1414 is being used by another queue manager, type another port number, such as `1416` or `1515`.
8. Accept the default Broker Domain name, Queue Manager name, and Database name values.

   If components with those names already exist, you can change the names of the components that the Getting Started wizard creates. Use alphanumeric characters (a-z, A-Z, 0-9) and underscores (_). The database name has a maximum limit of 8 characters.


WebSphere Business Integration Message Broker Getting Started Wizard

**Specify Broker Domain Details**

Enter the port number for the listener that will be created on the Queue Manager. The default port 1414 can be specified if not already in use.

Broker Domain name: LocalDomain

Queue Manager name: WBRK_QM

Queue Manager port:

Database name: WBRKCMDB

< Back
Next >
Finish
Cancel

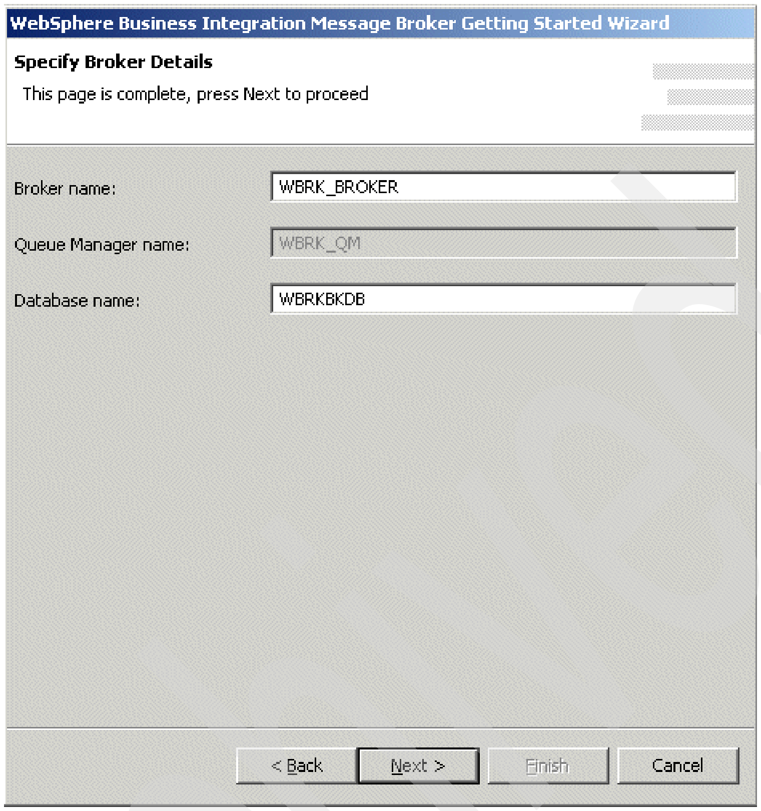
*Figure 3-23 Broker Domain Details page*

9. Click **Next**.

   The wizard checks whether a database with the specified name already exists on the computer. If it does exist, you are asked to confirm that you want to continue. If you continue, data in the existing database might be lost. Alternatively, type a new name in the Database name field.

10. Accept the default Broker name and Database name values (Figure 3-24 on page 62).

    If components with those names already exist, you can change the names of the components that the Getting Started wizard creates. Use alphanumeric characters (a-z, A-Z, 0-9) and underscores (_). The database name has a maximum limit of 8 characters.


WebSphere Business Integration Message Broker Getting Started Wizard

**Specify Broker Details**

This page is complete, press Next to proceed

Broker name: WBRK_BROKER

Queue Manager name: WBRK_QM

Database name: WBRKBKDB

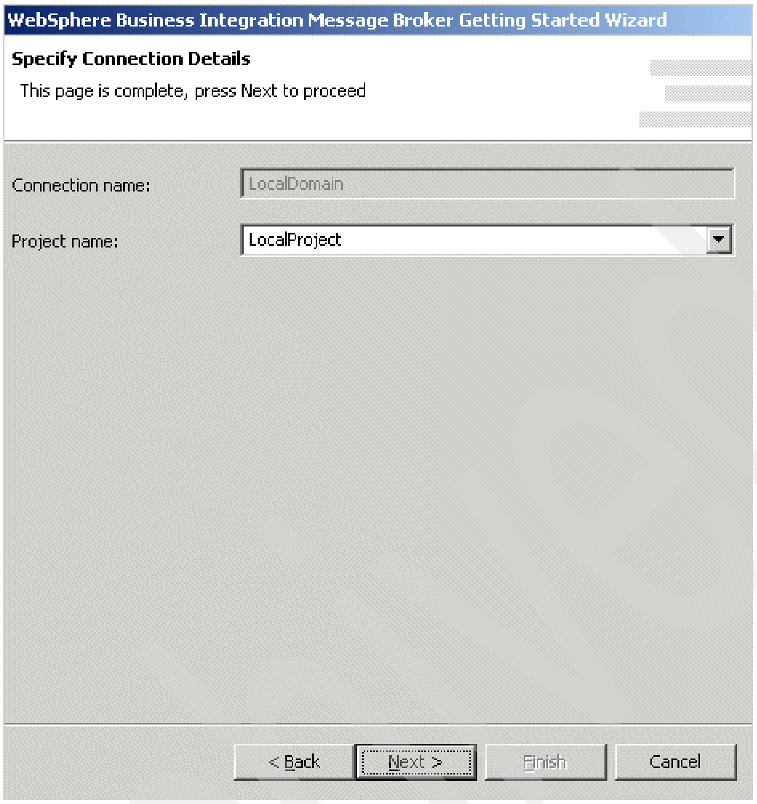
*Figure 3-24 Broker Details page*

11. Click **Next**.

    The wizard checks whether the a database with the specified name already exists on the computer. If it does exist, you are asked to confirm that you want to continue. If you continue, data in the existing database might be lost. Alternatively, type a new name in the Database name field. The wizard again checks whether the database specified for the broker already exists

12. Accept the default project name for the Server project that the wizard creates (Figure 3-25 on page 63). The connection name is not editable because you specified it previously, on the Broker Domain Details page.

    If one or more server projects already exist, select one of these projects from the list. If you are running the Getting Started wizard directly after installing the product, the only available option is `LocalProject`; however, you can edit this field if you prefer.


WebSphere Business Integration Message Broker Getting Started Wizard

Specify Connection Details

This page is complete, press Next to proceed

Connection name: LocalDomain

Project name: LocalProject

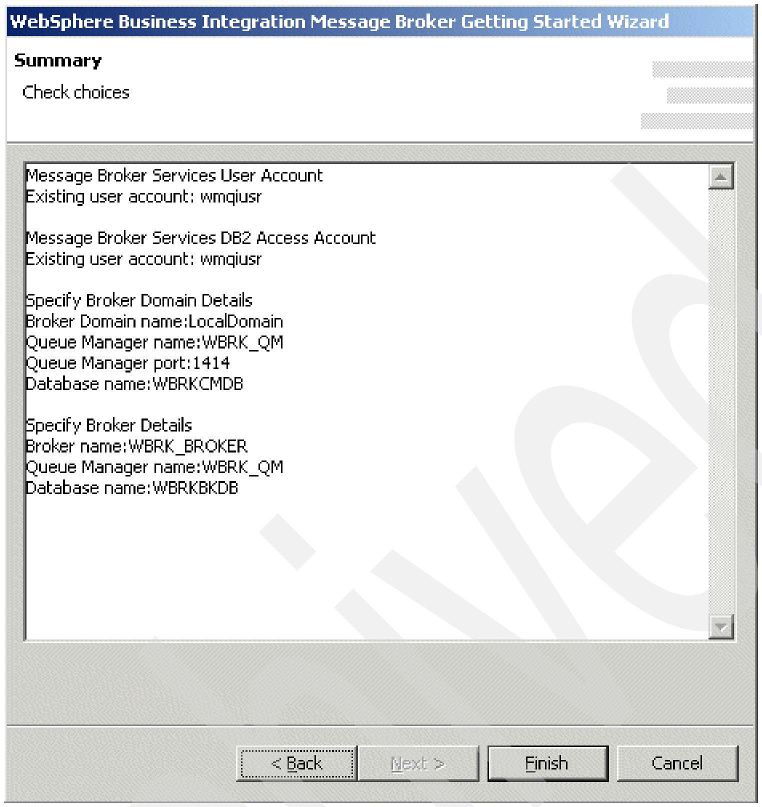
*Figure 3-25 Connection Details specification page*

13. Click **Next**.

    This opens a summary screen, which lists the choices that you made on previous pages of the wizard (Figure 3-26 on page 64).

    (Optional) To change any of these choices, click **Back** to return to the appropriate page of the wizard and make the changes.

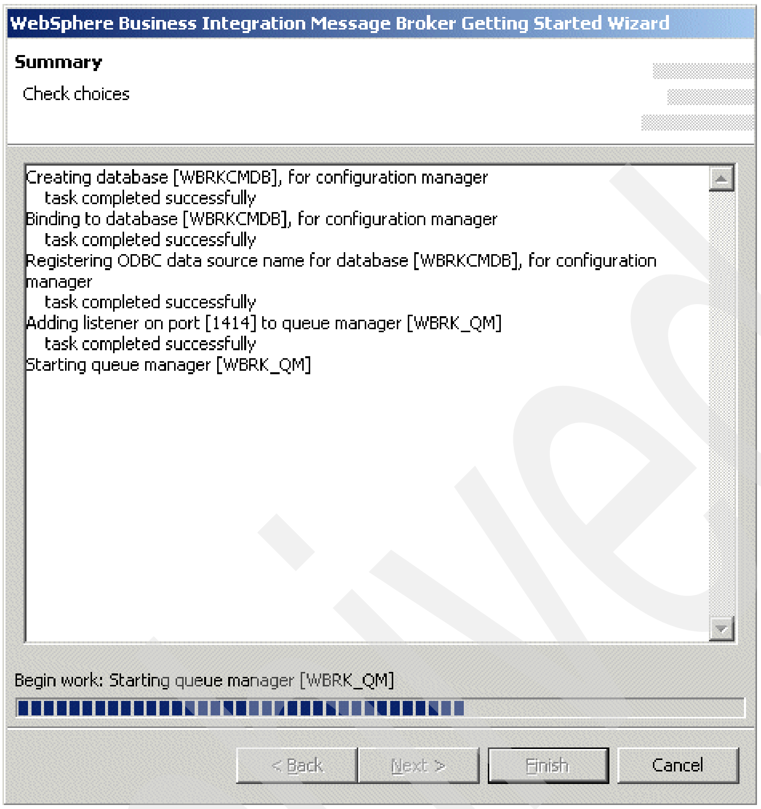
14. To start the configuration, click **Next**.


WebSphere Business Integration Message Broker Getting Started Wizard
Summary
Check choices
< Back
Next >
Finish
Cancel

```
Message Broker Services User Account
Existing user account: wmqiusr

Message Broker Services DB2 Access Account
Existing user account: wmqiusr

Specify Broker Domain Details
Broker Domain name:LocalDomain
Queue Manager name:WBRK_QM
Queue Manager port:1414
Database name:WBRKCMDB

Specify Broker Details
Broker name:WBRK_BROKER
Queue Manager name:WBRK_QM
Database name:WBRKBKDB
```

*Figure 3-26 The Getting Started Wizard Summary page*

The Getting Started wizard displays a progress bar (Figure 3-27 on page 65) as it creates the components that are listed in Table 3-3 on page 56.


WebSphere Business Integration Message Broker Getting Started Wizard
Summary
Check choices

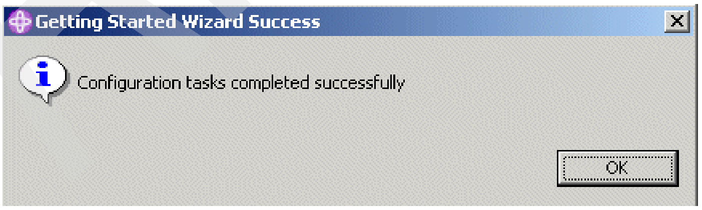
```
Creating database [WBRKCMDB], for configuration manager
   task completed successfully
Binding to database [WBRKCMDB], for configuration manager
   task completed successfully
Registering ODBC data source name for database [WBRKCMDB], for configuration
manager
   task completed successfully
Adding listener on port [1414] to queue manager [WBRK_QM]
   task completed successfully
Starting queue manager [WBRK_QM]
```

Begin work: Starting queue manager [WBRK_QM]

*Figure 3-27 The Getting Started Wizard in action*

15. When the configuration is complete, the wizard displays a message:

   – If the message says that the configuration was completed successfully and that all components were created, click **OK** (Figure 3-28). A second confirmation message is displayed. Click **OK** to close the Getting Started wizard.



Configuration tasks completed successfully

*Figure 3-28 Successful default configuration created*

– If the message says that the configuration failed due to an error, the error message contains details of the error. To diagnose and resolve the problem, and to read about commonly occurring problems, see 6.4.1, “Getting Started wizard fails” on page 207.

## 3.6 Verifying the installation

A range of samples is provided with WebSphere Business Integration Message Brokers to test the system configuration and to learn more about the capabilities of the product. To run any of the samples, you must have configured WebSphere Business Integration Message Brokers.

To verify the installation of Message Broker, run the Getting Started samples.

To verify the installation of Event Broker, run the Soccer Results sample. The Getting Started samples use features that are not available in Event Broker so they are not supplied with it.

For more information about preparing and running the Soccer Results sample and the other samples, see the Samples Gallery in the product documentation (click **Help** → **Help Contents**).

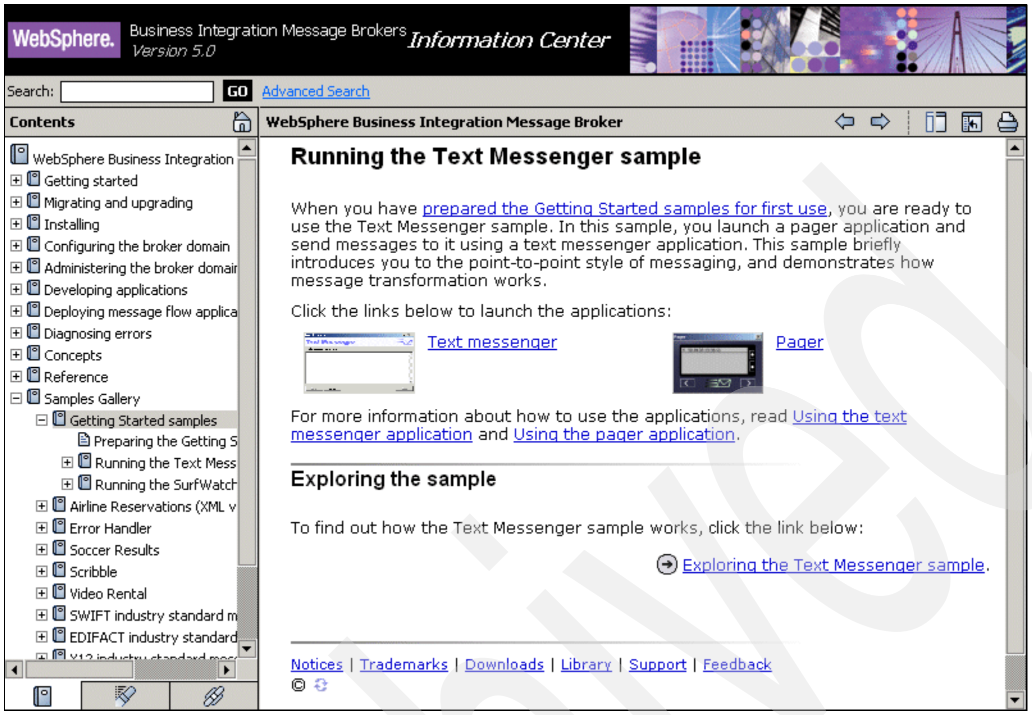
### 3.6.1 Preparing to run the Getting Started samples

The instructions in this section describe how to prepare the Getting Started samples for first use. (If you have installed Event Broker, prepare the Soccer Results sample instead.) For more help, see the Samples Gallery in the product documentation (click **Help** → **Help Contents**).

To import and deploy the Getting Started samples:

1. On the Message Brokers Toolkit welcome page (Figure 3-20 on page 57), click the **Getting Started samples** link in the Getting Started section.

   The documentation for the Getting Started samples opens in a separate window (Figure 3-29 on page 67).


WebSphere. Business Integration Message Brokers Version 5.0 Information Center
Search: GO Advanced Search
Contents
WebSphere Business Integration Message Broker
WebSphere Business Integration
Getting started
Migrating and upgrading
Installing
Configuring the broker domain
Administering the broker domair
Developing applications
Deploying message flow applica
Diagnosing errors
Concepts
Reference
Samples Gallery
Getting Started samples
Preparing the Getting S
Running the Text Mess
Running the SurfWatch
Airline Reservations (XML v
Error Handler
Soccer Results
Scribble
Video Rental
SWIFT industry standard m
EDIFACT industry standard

## Running the Text Messenger sample

When you have prepared the Getting Started samples for first use, you are ready to use the Text Messenger sample. In this sample, you launch a pager application and send messages to it using a text messenger application. This sample briefly introduces you to the point-to-point style of messaging, and demonstrates how message transformation works.

Click the links below to launch the applications:

Text messenger Pager

For more information about how to use the applications, read Using the text messenger application and Using the pager application.

### Exploring the sample

To find out how the Text Messenger sample works, click the link below:

Exploring the Text Messenger sample.

Notices | Trademarks | Downloads | Library | Support | Feedback

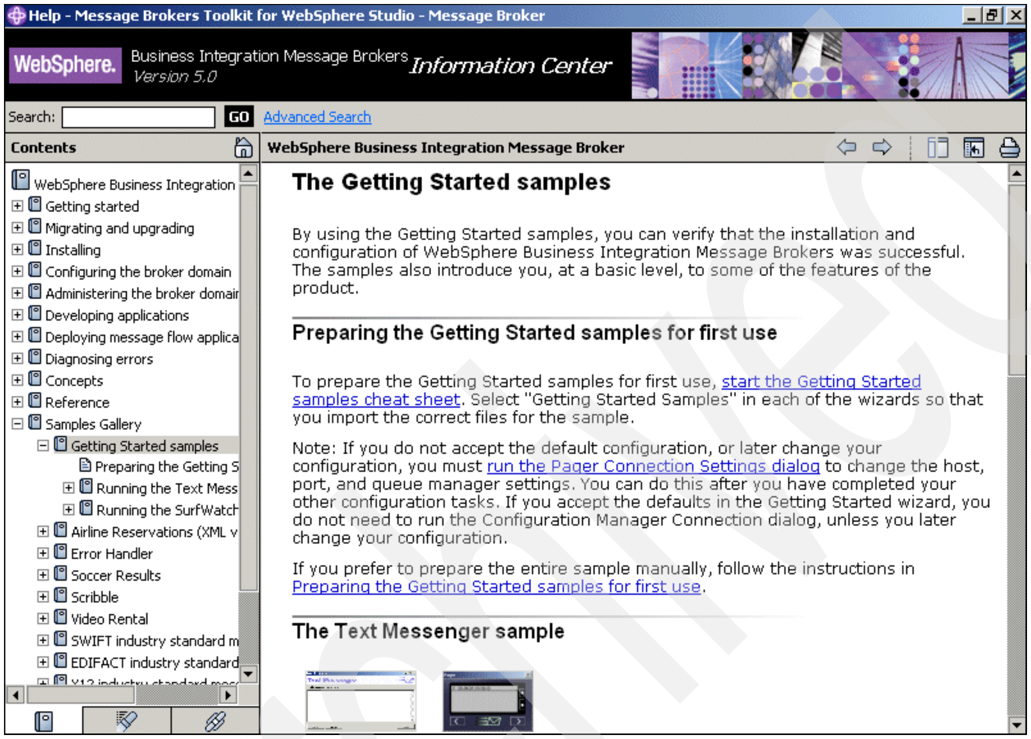
*Figure 3-29 The Getting Started samples documentation in the Samples Gallery*

2. In the Contents view (on the left), select **Samples Gallery** → **Getting Started samples** → **Preparing the Getting Started samples for first use**.

   The instructions for preparing to run the Getting Started samples is displayed (Figure 3-30 on page 68).

3. Click the first link on the page, which is labeled **Start the Getting Started samples cheat sheet**, then minimize the Help window.

   The cheat sheet opens in the Message Brokers Toolkit *behind* the Help window.



Help - Message Brokers Toolkit for WebSphere Studio - Message Broker
WebSphere. Business Integration Message Brokers Version 5.0 Information Center
Search: GO Advanced Search
Contents
WebSphere Business Integration Message Broker
WebSphere Business Integration
Getting started
Migrating and upgrading
Installing
Configuring the broker domain
Administering the broker domair
Developing applications
Deploying message flow applica
Diagnosing errors
Concepts
Reference
Samples Gallery
Getting Started samples
Preparing the Getting S
Running the Text Mess
Running the SurfWatch
Airline Reservations (XML v
Error Handler
Soccer Results
Scribble
Video Rental
SWIFT industry standard m
EDIFACT industry standard

**The Getting Started samples**

By using the Getting Started samples, you can verify that the installation and configuration of WebSphere Business Integration Message Brokers was successful. The samples also introduce you, at a basic level, to some of the features of the product.

**Preparing the Getting Started samples for first use**

To prepare the Getting Started samples for first use, start the Getting Started samples cheat sheet. Select "Getting Started Samples" in each of the wizards so that you import the correct files for the sample.

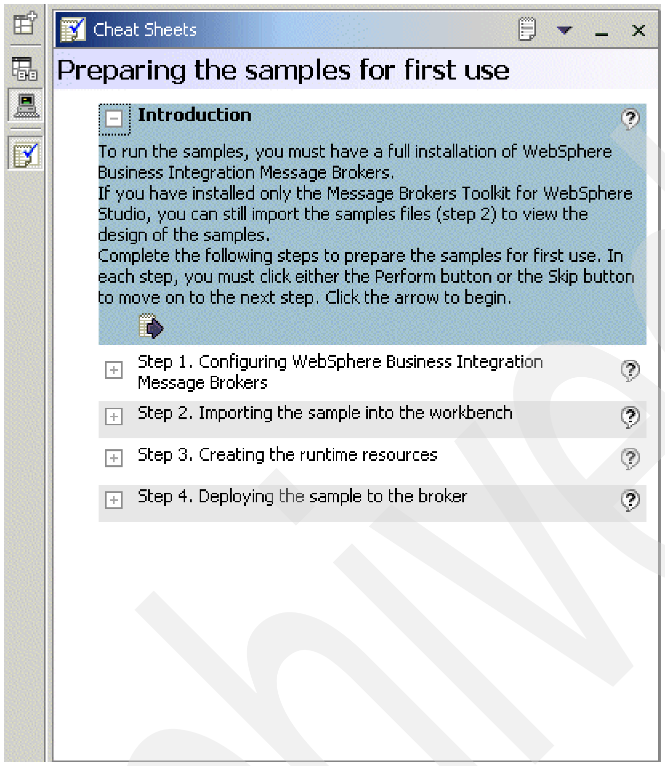
Note: If you do not accept the default configuration, or later change your configuration, you must run the Pager Connection Settings dialog to change the host, port, and queue manager settings. You can do this after you have completed your other configuration tasks. If you accept the defaults in the Getting Started wizard, you do not need to run the Configuration Manager Connection dialog, unless you later change your configuration.

If you prefer to prepare the entire sample manually, follow the instructions in Preparing the Getting Started samples for first use.

**The Text Messenger sample**

*Figure 3-30 The Getting Started samples documentation*

Figure 3-31 shows the cheat sheet in the Message Brokers Toolkit.



Preparing the samples for first use

**Introduction**

To run the samples, you must have a full installation of WebSphere Business Integration Message Brokers.
If you have installed only the Message Brokers Toolkit for WebSphere Studio, you can still import the samples files (step 2) to view the design of the samples.
Complete the following steps to prepare the samples for first use. In each step, you must click either the Perform button or the Skip button to move on to the next step. Click the arrow to begin.

Step 1. Configuring WebSphere Business Integration Message Brokers

Step 2. Importing the sample into the workbench

Step 3. Creating the runtime resources

Step 4. Deploying the sample to the broker

*Figure 3-31 Preparing the samples for first use cheat sheet*

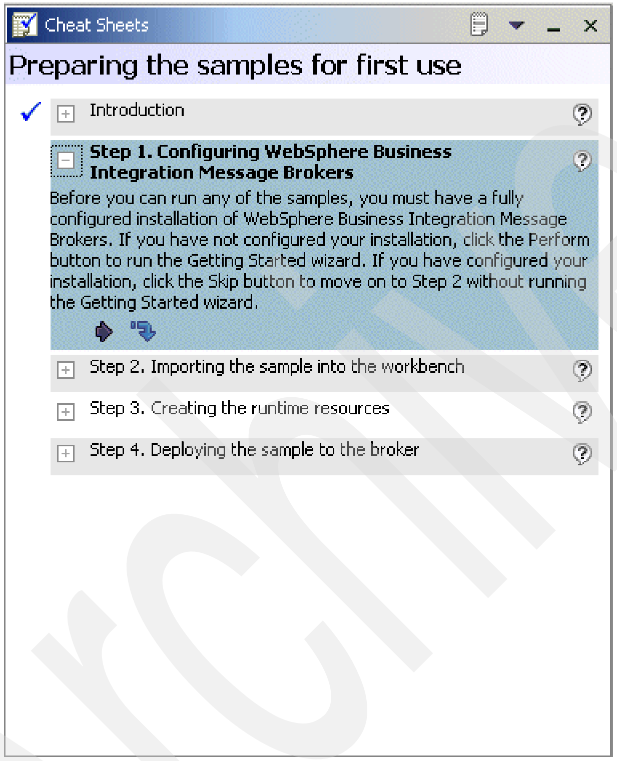
The following sections describe the steps for using the cheat sheet to prepare the Getting Started samples for first use:

- Configuring the installation of WebSphere Business Integration Message Broker
- Importing the Getting Started samples files into the Message Brokers Toolkit
- Creating the runtime resources, such as queues
- Deploying the sample message flow application to the broker

## Configuring the installation

You already configured the installation of WebSphere Business Integration Message Broker in 3.4, "Installing product fix packs" on page 52 of this chapter, so skip Step 1 of the cheat sheet.

1. In the **Introduction** section of the cheat sheet, click the arrow. If you hover the mouse pointer over the icon, it is labeled "Click to begin." Step 1 of the cheat sheet is expanded (Figure 3-32).



Preparing the samples for first use

Introduction

**Step 1. Configuring WebSphere Business Integration Message Brokers**

Before you can run any of the samples, you must have a fully configured installation of WebSphere Business Integration Message Brokers. If you have not configured your installation, click the Perform button to run the Getting Started wizard. If you have configured your installation, click the Skip button to move on to Step 2 without running the Getting Started wizard.

Step 2. Importing the sample into the workbench

Step 3. Creating the runtime resources

Step 4. Deploying the sample to the broker

*Figure 3-32 Step 1: Configuring WebSphere Business Integration Message Brokers*

2. Do *not* perform Step 1 of the cheat sheet. Step 1 enables you to run the Getting Started wizard to create a default configuration. You have already configured the system.

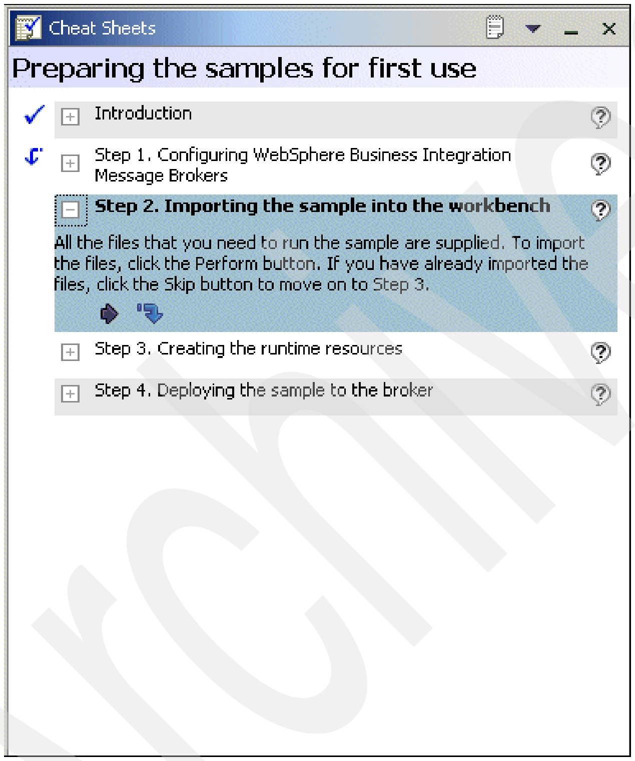
   Click the downward-pointing arrow to continue to Step 2. Step 2 expands and a large, downward-pointing arrow is displayed alongside Step 1 to show that you skipped the tasks in the step (Figure 3-33 on page 71).

## Importing the Getting Started samples files

The Getting Started samples are supplied in plug-ins in the Message Brokers Toolkit. Import the Getting Started samples files into the Message Brokers Toolkit workspace so that you can deploy them to the broker.

To import the Getting Started samples:

1. In Step 2 of the cheat sheet, click the right-facing arrow (if you hover the mouse pointer over the arrow, it is labeled "Perform."



Cheat Sheets

Preparing the samples for first use

Introduction

Step 1. Configuring WebSphere Business Integration Message Brokers

**Step 2. Importing the sample into the workbench**

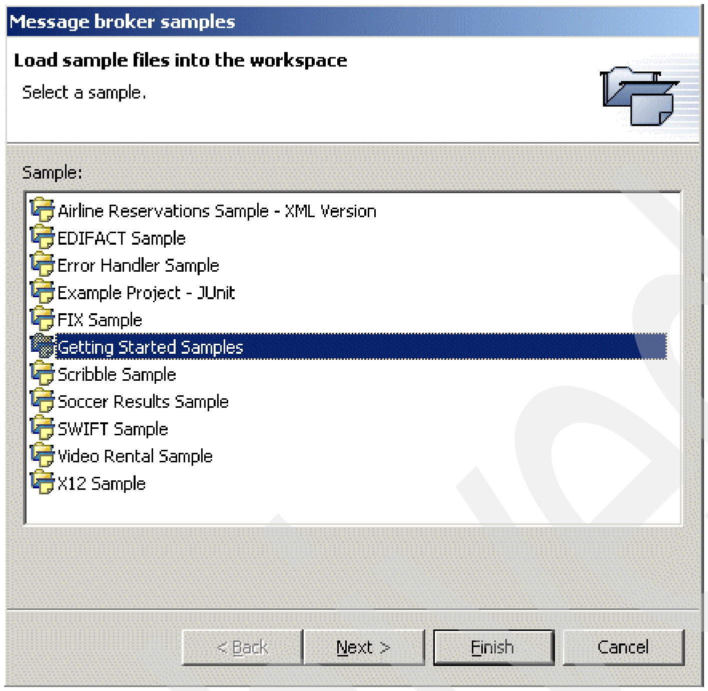
All the files that you need to run the sample are supplied. To import the files, click the Perform button. If you have already imported the files, click the Skip button to move on to Step 3.

Step 3. Creating the runtime resources

Step 4. Deploying the sample to the broker

*Figure 3-33 Step 2: Importing samples into the Message Brokers Toolkit*

2. A wizard opens for importing the samples into the Message Brokers Toolkit, as shown in Figure 3-34 on page 72.


Message broker samples

**Load sample files into the workspace**

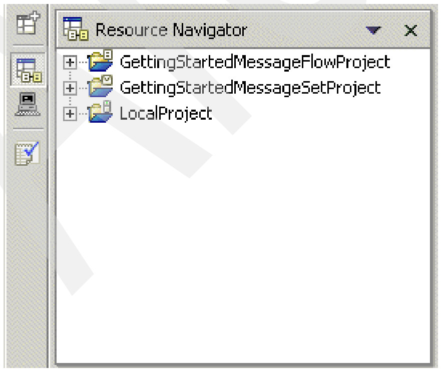
Select a sample.

Sample:

- Airline Reservations Sample - XML Version
- EDIFACT Sample
- Error Handler Sample
- Example Project - JUnit
- FIX Sample
- Getting Started Samples
- Scribble Sample
- Soccer Results Sample
- SWIFT Sample
- Video Rental Sample
- X12 Sample

*Figure 3-34 Importing Getting Started samples into the Message Brokers Toolkit*

3. Click **Getting Started Samples**, then click **Finish**. (The other pages in the wizard display the names of the projects that are imported.) Do not enter any more information in this wizard.
4. Minimize the cheat sheet, then view the contents of the Resource Navigator view (in the top-left corner of the Message Brokers Toolkit, Figure 3-35.


Resource Navigator

- GettingStartedMessageFlowProject
- GettingStartedMessageSetProject
- LocalProject

*Figure 3-35 Resource Navigator view after importing Getting Started samples*

5. To continue preparing the Getting Started samples for first use, click the cheat sheet icon on the left-side toolbar, as shown in Figure 3-35 on page 72.

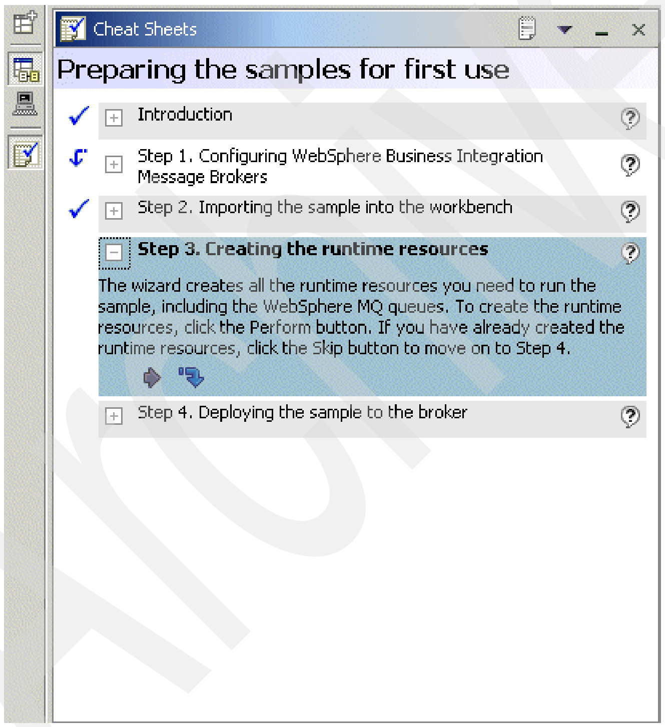
   A large check mark is now displayed alongside Step 2 of the cheat sheet to show that you have completed the tasks in the step. Step 3 is expanded, as shown in Figure 3-36.

## Creating the runtime resources

The message flows in the Getting Started samples interact with WebSphere MQ queues. Step 3 of the cheat sheet creates the queues, the runtime resources, that are required to run the Getting Started samples.

To create the runtime resources:

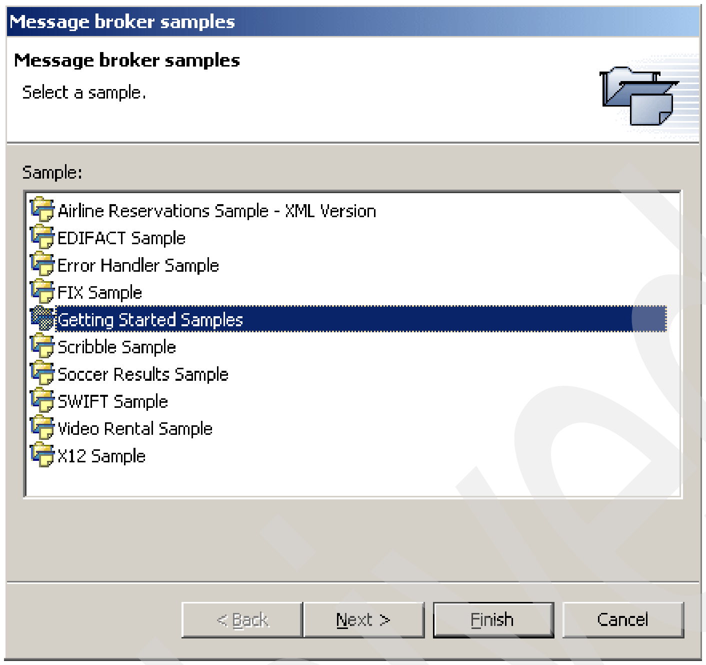
1. In Step 3 of the cheat sheet, click the Perform arrow.


Cheat Sheets
Preparing the samples for first use
Introduction
Step 1. Configuring WebSphere Business Integration Message Brokers
Step 2. Importing the sample into the workbench
Step 3. Creating the runtime resources
Step 4. Deploying the sample to the broker

The wizard creates all the runtime resources you need to run the sample, including the WebSphere MQ queues. To create the runtime resources, click the Perform button. If you have already created the runtime resources, click the Skip button to move on to Step 4.

*Figure 3-36 Step 3: Creating the runtime resources*

   A wizard opens to create the runtime resources for the Getting Started samples, as shown in Figure 3-37 on page 74.


Message broker samples

Message broker samples

Select a sample.

Sample:

- Airline Reservations Sample - XML Version
- EDIFACT Sample
- Error Handler Sample
- FIX Sample
- Getting Started Samples
- Scribble Sample
- Soccer Results Sample
- SWIFT Sample
- Video Rental Sample
- X12 Sample

*Figure 3-37 Creating the runtime resources for the Getting Started samples*

2. In the wizard, click **Getting Started Samples**, then click **Finish**. (The other pages in the wizard simply display the names of the queues that were created.) There is no need to enter any more information in this wizard.

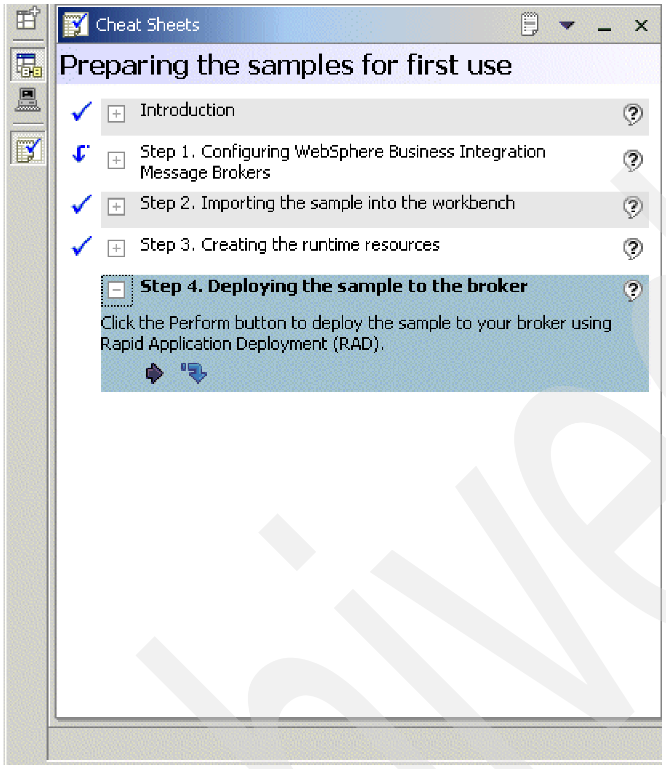
   The WebSphere MQ queues are created and a message is displayed to confirm that the queues were created successfully.

3. Click **Finish**. A successful-creation message appears. Click **OK** to close it.

   A large check mark is now displayed alongside Step 3 of the cheat sheet to show that you have completed the tasks in the step. Step 4 is expanded, as shown in Figure 3-38 on page 75.

### Deploying the sample to the broker

The final step of the cheat sheet guides you through the task of deploying the Getting Started samples' message flow and message set projects to the broker.



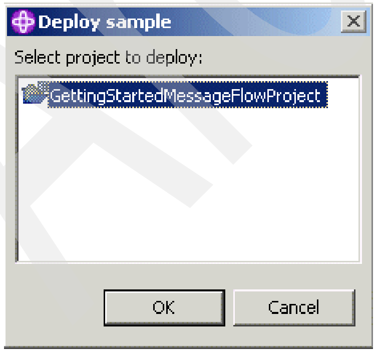
Preparing the samples for first use

- Introduction
- Step 1. Configuring WebSphere Business Integration Message Brokers
- Step 2. Importing the sample into the workbench
- Step 3. Creating the runtime resources
- **Step 4. Deploying the sample to the broker**

Click the Perform button to deploy the sample to your broker using Rapid Application Deployment (RAD).

*Figure 3-38 Step 4: Preparing the samples for first use*

1. In Step 4 of the cheat sheet, click the Perform arrow. The Deploy sample dialog opens, as shown in Figure 3-39.



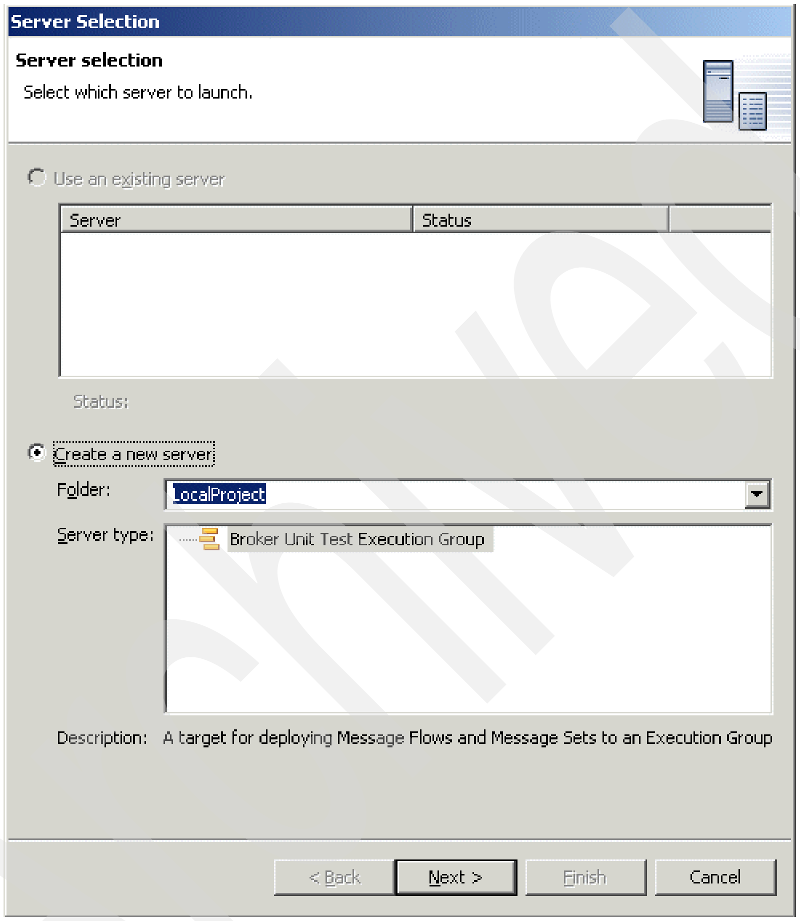
Select project to deploy:

GettingStartedMessageFlowProject

*Figure 3-39 Deploy sample dialog*

2. In the Deploy sample dialog, click **GettingStartedMessageFlowProject**, then click **OK**.

   The Deploy sample dialog closes and the Server Selection wizard opens, as shown Figure 3-40.



Server Selection

**Server selection**

Select which server to launch.

Use an existing server

| Server | Status | |
|---|---|---|

Status:

Create a new server [✓]

Folder: LocalProject

Server type: Broker Unit Test Execution Group

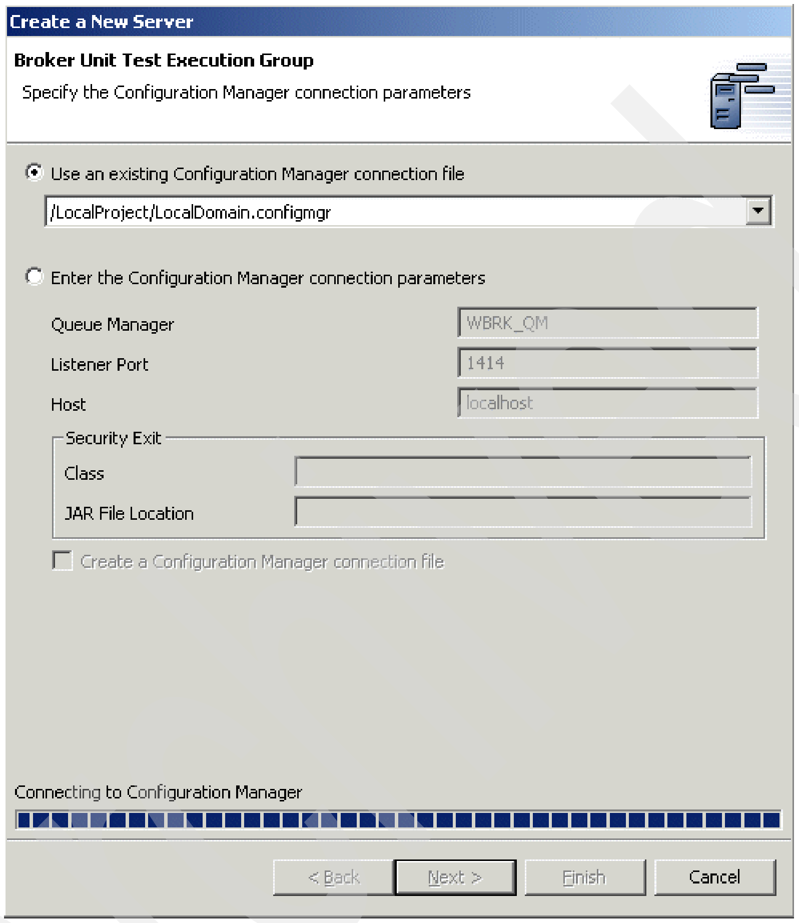
Description: A target for deploying Message Flows and Message Sets to an Execution Group

*Figure 3-40 Server Selection wizard*

*Server* is the term used in the Message Brokers Toolkit for the configuration details of a Rapid Application Development (RAD) deploy. A RAD deploy is a quick deployment method that you can use when developing message flow applications. For more about RAD deploy, see 5.2, "Deploying message flow applications" on page 147.

3. In the Server Selection wizard, select **Create a new server**, then click **Next**.

4. Click **Next** to accept the default Configuration Manager connection parameters shown in Figure 3-41.


Create a New Server

Broker Unit Test Execution Group

Specify the Configuration Manager connection parameters

(●) Use an existing Configuration Manager connection file

/LocalProject/LocalDomain.configmgr

( ) Enter the Configuration Manager connection parameters

| | |
|---|---|
| Queue Manager | WBRK_QM |
| Listener Port | 1414 |
| Host | localhost |

Security Exit

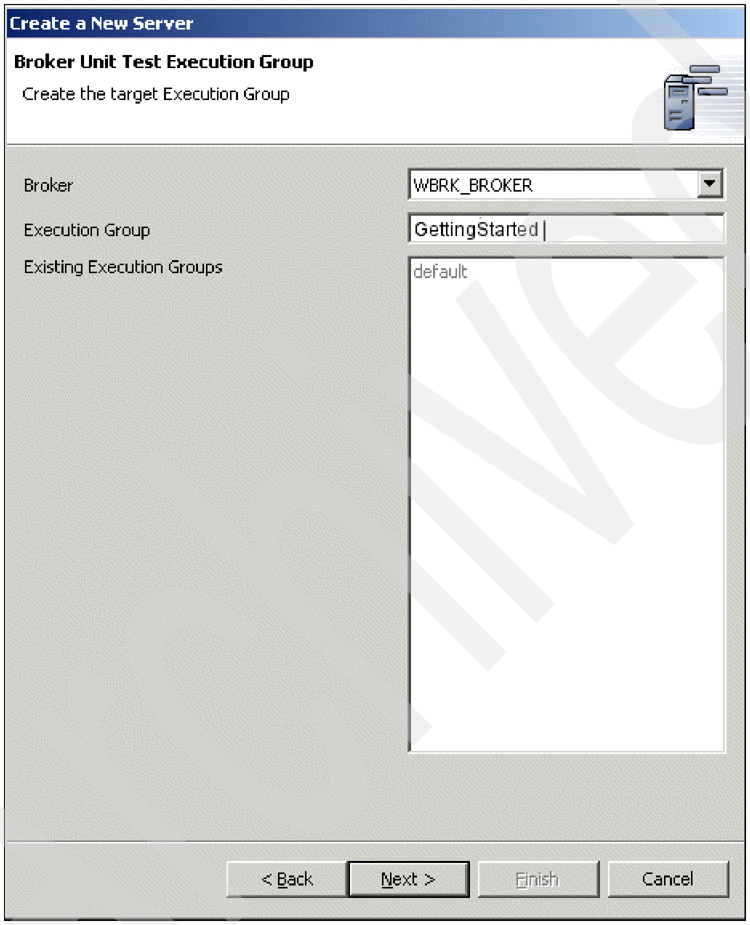
| | |
|---|---|
| Class | |
| JAR File Location | |

☐ Create a Configuration Manager connection file

Connecting to Configuration Manager

Figure 3-41 Create server connection

5. In the Execution Group field, clear the default value (the user ID that you are logged on as), then type `GettingStarted` to create a new execution group called GettingStarted (Example 3-42).
6. Ensure that the correct broker name is displayed in the Broker field. Click **Next**.


Create a New Server

**Broker Unit Test Execution Group**

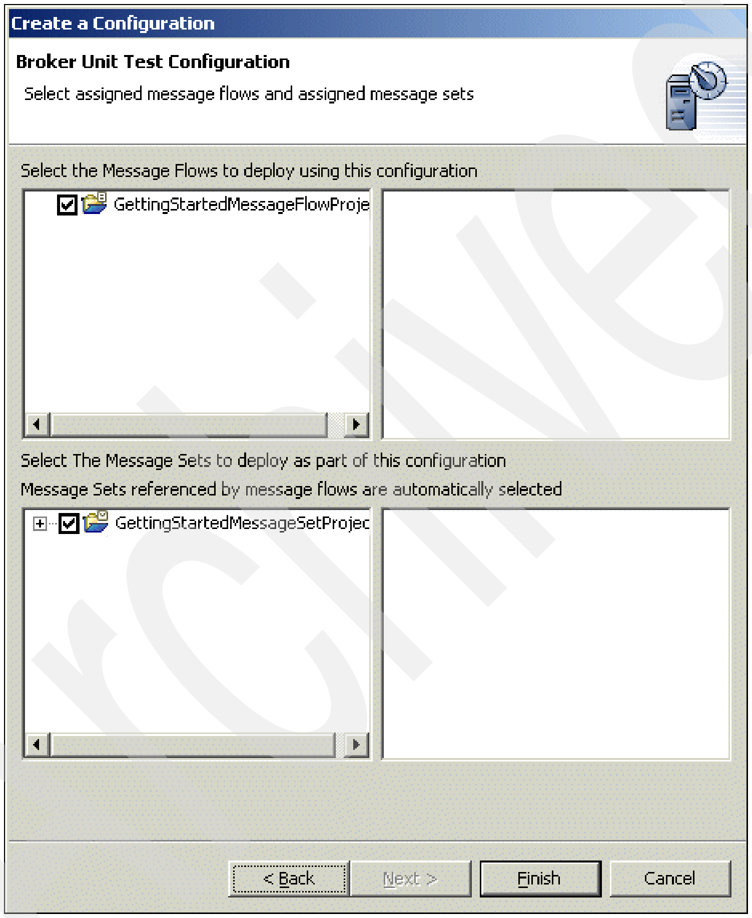
Create the target Execution Group

| | |
|---|---|
| Broker | WBRK_BROKER |
| Execution Group | GettingStarted |
| Existing Execution Groups | default |

*Figure 3-42 Creating the execution group called GettingStarted*

7. Select the check box next to **GettingStartedMessageFlowProject** (Figure 3-43). This automatically selects the check box next to **GettingStartedMessageSetProject** because the message flow in GettingStartedMessageFlowProject does not work without the message set in GettingStartedMessageSetProject.

   See 5.2.2, "Setting up a RAD deploy configuration" on page 148 for more information about project references.


Create a Configuration

Broker Unit Test Configuration

Select assigned message flows and assigned message sets

Select the Message Flows to deploy using this configuration

☑ GettingStartedMessageFlowProje

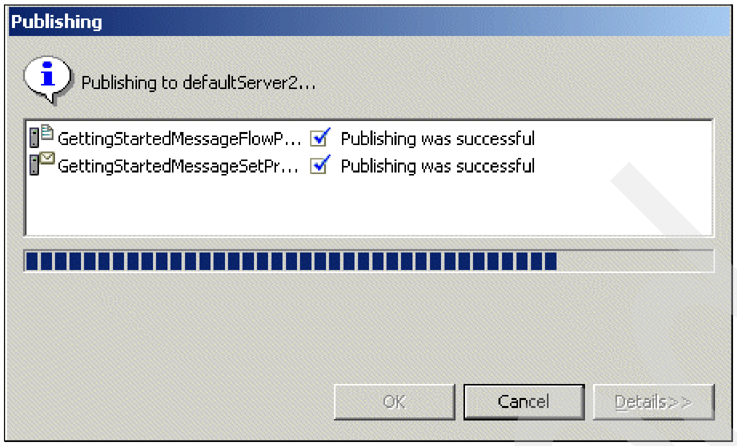
Select The Message Sets to deploy as part of this configuration

Message Sets referenced by message flows are automatically selected

☑ GettingStartedMessageSetProjec

*Figure 3-43 Broker Unit Test Configuration*

8. Click **Finish**. The sample message flows and message set are deployed to the GettingStarted execution group. In a RAD deploy, this process is known as *publishing* the message flow and message set to the execution group, GettingStarted, that you configured in the broker. A message displays the progress of the publishing, as shown in Figure 3-44 on page 80.



GettingStartedMessageFlowP... Publishing was successful
GettingStartedMessageSetPr... Publishing was successful

*Figure 3-44 Publishing to server*

When the message flow and message set are successfully deployed, you might see an error message that says `Cannot start the server because it is already running`. You can safely ignore this message. The error occurs because Message Broker attempts to publish the message flow and message set before the execution group has started. Typically, the deployment succeeds anyway. If the deployment fails, return to Step 4 of the cheat sheet and click the Perform button to rerun the wizard. Use the server that you just created.

You have now prepared the Getting Started samples for first use. You can now run the Getting Started samples.

## Running the Getting Started samples

The Getting Started samples are supplied with graphical interfaces so that you can run them easily.

To run the Getting Started samples:

1. Close the cheat sheet.
2. On the Message Brokers Toolkit welcome page, click the **Getting Started samples** link to open the Getting Started samples documentation.

   The page that opens is about running the Text Messenger sample.

3. Follow the instructions in the Getting Started samples documentation to start the Text Messenger and Pager graphical interfaces, and to run the samples.

The Text Messenger sample demonstrates the point-to-point style of messagin.g (see Chapter 1, “Introduction and overview” on page 1 for a brief explanation of point-to-point messaging.)

To learn about the publish/subscribe style of messaging, also run the SurfWatch sample.

To prepare and run the other samples that are supplied with WebSphere Business Integration Message Brokers, refer to the guidance in the Samples Gallery in the product documentation.

The following chapters in this book explore how to develop simple message flow applications, how to manually create broker domains, and how to diagnose problems in WebSphere Business Integration Message Brokers.

4

# Application development

This chapter describes how to develop message flow and message set applications in the Message Brokers Toolkit for WebSphere Studio using WebSphere Business Integration Message Broker.

This chapter covers:

- Developing a simple message flow
- Developing a message set
- Programming more complex message flow applications

## 4.1 Developing message flow applications

A message flow application is a program that processes messages in the broker. The first part of this chapter describes how to create a simple message flow that takes an Extensible Markup Language (XML) message from one WebSphere MQ queue, copies the contents of the message to an output message, and passes the output message to a second WebSphere MQ queue. The XML message that the message flow processes is self-defining. This means that there is no external definition of how the message content is structured. All of the information about the structure of the message content is in the XML tagging of the message elements. Example 4-1 shows the content of the XML message that is used to test the message flows that are described in this chapter.

*Example 4-1 Content of XML message used to test the message flow in this chapter*

```
<BookStoreOrder>
   <CustomerDetails>
      <FirstName>Sandra</FirstName>
      <LastName>Postlethwaite</LastName>
      <Address>7 Mycenae Street</Address>
      <ZipCode>SO21 2JN</ZipCode>
   </CustomerDetails>
   <Book>
      <Title>Sitting Pretty</Title>
      <Author>Tim Gordon</Author>
      <Edition>1</Edition>
      <Date>1976-05-23T01:00:00</Date>
      <Publisher>Hamble and Bursledon</Publisher>
      <ISBN>FT03449302 0</ISBN>
      <Genre>Thriller</Genre>
   </Book>
   <BookOrder>
      <Format>Hardback</Format>
      <Price>15.00</Price>
   </BookOrder>
</BookStoreOrder>
```

The message flow application that is described in this chapter is based around the scenario of an online bookstore. The XML message in Figure 4-1 shows the order details of a customer called Sandra Postlethwaite. The message contains the customer's personal details and details of the book she has ordered, such as the title of the book, `Sitting Pretty`, the book's author, `Tim Gordon`, the price of the book, `15.00`, and so on.

The second part of this chapter describes how to modify the BookstoreFlow message flow so that the structure of messages that the message flow processes is defined externally in a message set.

The third part of this chapter describes the different manipulations and transformations that you can program a message flow application to perform using the built-in nodes in WebSphere Business Integration Message Broker.

The message flow application that is described in this chapter uses the point-to-point style of messaging and can be developed only in Message Broker. Many of the nodes discussed in this chapter are not available in WebSphere Business Integration Event Broker. Although the message flow application described here cannot be developed using Event Broker, the principles of creating, deploying, and testing a message flow are the same. Note, however, that you cannot develop message sets in Event Broker.

## 4.2 Developing a simple message flow

A message flow is a series of steps that route, transform, or manipulate a message when it arrives at the broker from an application before the broker passes the message to another application. Program a message flow by connecting a sequence of nodes, configuring the nodes' properties, and writing Extended Structured Query Language (ESQL) to manipulate the data, route the message to another message flow, or simply alter the message in some way.

The following sections describe how to create a simple message flow in the Message Brokers Toolkit using the Message Flow Editor and the ESQL editor. Both editors belong to the Broker Application Development perspective. The Broker Application Development perspective is the default perspective that opens when you start the Message Brokers Toolkit.

This chapter also describes how to test the message flow by deploying it to the broker and putting a test message through it using the Enqueue Message editor, which belongs to the Broker Domain Administration perspective.

The instructions in this chapter assume that you have run the Getting Started wizard to create a default configuration, and that the broker (WBRK_BROKER) and the Configuration Manager are running.

To start the broker, enter the following command at a command prompt:

```
mqsistart wbrk_broker
```

To start the Configuration Manager, enter the following command:

```
mqsistart configmgr
```

## 4.2.1 Building the message flow

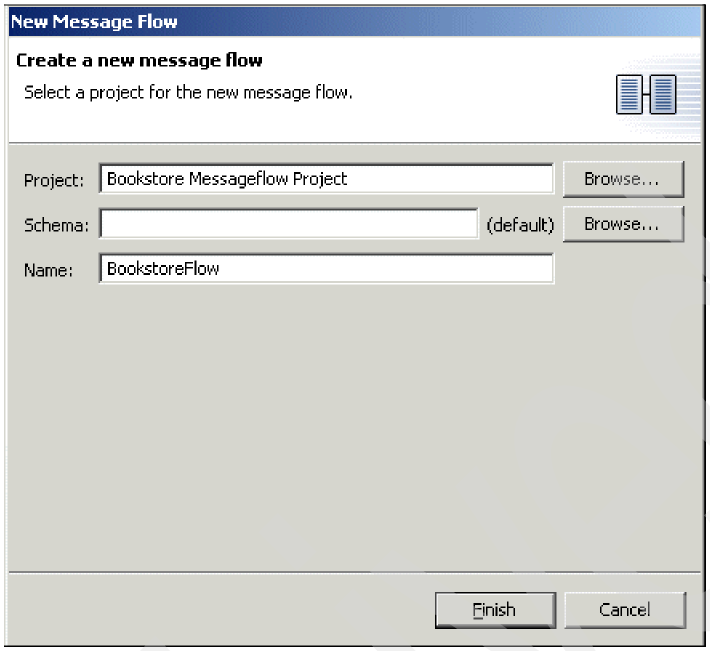
The message flow that is described in this chapter consists of three nodes:

- An MQInput node (to get the input message from the WebSphere MQ queue)
- A Compute node (to manipulate the message)
- An MQOutput node (to put the output message onto a second WebSphere MQ queue)

### Creating the BookstoreFlow message flow

Before you can start adding and configuring message flow nodes, create the files in which the message flow is stored:

1. In the Broker Application Development perspective, create a message flow project called Bookstore Messageflow Project:
   a. Click **File** → **New** → **Message Flow Project**.
   b. In the Project Name field, type `Bookstore Messageflow Project`, then click **Finish**. A new project called Bookstore Messageflow Project is displayed in the Resource Navigator viewat the top-left of the Message Brokers Toolkit window.
2. Create the BookstoreFlow message flow in the Bookstore Messageflow Project:
   a. In the Resource Navigator view, click the **Bookstore Messageflow Project** to highlight it.
   b. Click **File** → **New** → **Message Flow**.
   c. Ensure that the Project field contains `Bookstore Messageflow Project` (Figure 4-1 on page 87).


New Message Flow

Create a new message flow

Select a project for the new message flow.

Project: Bookstore Messageflow Project

Schema: (default)

Name: BookstoreFlow

*Figure 4-1 Creating the Bookstore Messageflow Project*

d. Leave the Schema field empty so that the message flow is created in the default schema.

e. In the Name field, type `BookstoreFlow`, then click **Finish**.

In the Resource Navigator view, a file called BookstoreFlow.msgflow is now displayed in the Default schema of the Bookstore Messageflow Project. (For more information about schemas, see the product documentation: **Concepts** → **Message flows** → **Broker schema**.) The BookstoreFlow.msgflow file opens automatically in the Message Flow Editor, the tabbed area to the right of the Resource Navigator view.

## Adding and connecting the nodes

On the left side of the Message Flow Editor is the Node palette, a list of all nodes that are available for use in the message flow (Figure 4-2).

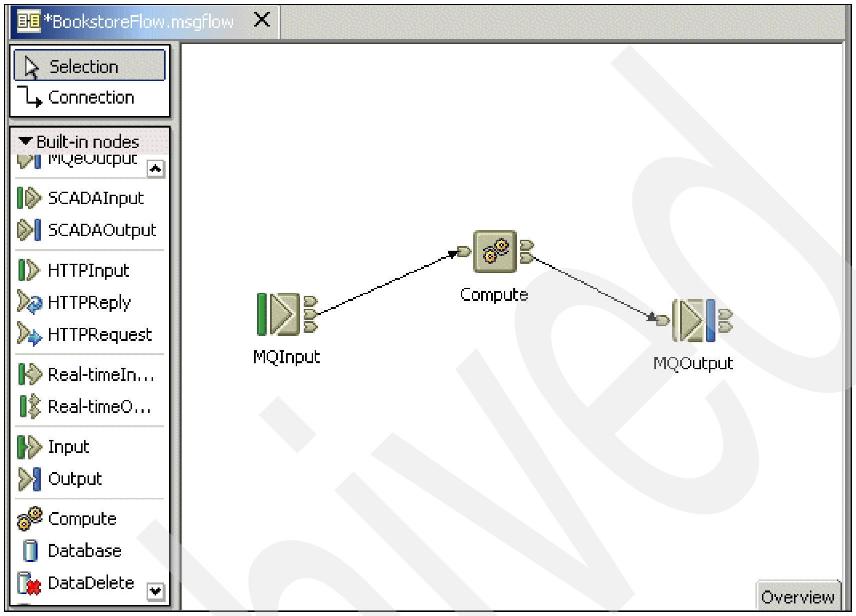


*Figure 4-2 BookstoreFlow message flow nodes*

To build the message flow shown in Figure 4-2:

1. Make sure that the Selection button (above the Node palette) is highlighted so that you can select nodes from the Node palette.
2. Click the **MQInput** node to select it from the Node palette, then click somewhere on the canvas (the white area to the right of the Node palette) to add the node to the message flow.
3. Repeat steps 1 and 2 to add a Compute node and an MQOutput node to the canvas.

To define the order that the nodes process an input message, connect them as shown in Figure 4-2. Most, though not all, nodes have at least one terminal with which to connect them to other nodes.

The MQInput node has three terminals. Hover the mouse pointer over each terminal to highlight it and to display its label. The terminals on the MQInput node are labeled Failure, Out, and Catch.

1. Click the **Connection** button (above the Node palette) so that you can connect the nodes together.
2. Click the **Out** terminal of the MQInput node. Notice that a thin black line with an arrow head appears. Move the mouse pointer and click the **In** terminal of the Compute node so that the arrow connects the MQInput node to the Compute node.
3. Connect the Out terminal of the Compute node to the In terminal of the MQOutput node.

   The canvas should look similar to the canvas shown in Figure 4-2 on page 88.

## Saving and validating the message flow

To save the BookstoreFlow message flow:

Click **File** → **Save *BookstoreFlow.msgflow***.

When saving a message flow file, the Message Brokers Toolkit validates the message flow. BookstoreFlow message flow has two errors, as shown in Figure 4-3, which are indicated by a small white cross on a red background on the MQInput and Compute nodes. The BookstoreFlow message flow files and folders in the Resource Navigator view are also highlighted with crosses.

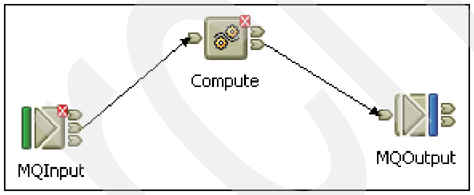


*Figure 4-3 The BookstoreFlow message flow validated*

A brief description of each error is given in the Tasks view below the Message Flow Editor:

- ► The error in the MQInput node is because you have not entered the name of the input queue from which the MQInput node takes input messages.
- ► The error in the Compute node is because you have not created the ESQL module that defines how the Compute node should process input messages.

The following sections of this chapter describe how to fix the errors by configuring the nodes.

## 4.2.2 Configuring the message flow

In the BookstoreFlow message flow, the MQInput node takes an input message from a WebSphere MQ local queue called IN. After the Compute node has processed the message, the MQOutput node puts the output message to a WebSphere MQ local queue called OUT. In the BookstoreFlow message flow, when the Compute node receives the input message from the MQInput node, the Compute node builds an output message that contains a copy of the header and content of the input message.

To configure the message flow, create the two WebSphere MQ local queues, IN and OUT, and create the ESQL module that processes the message in the Compute node.

### Creating the WebSphere MQ local queues

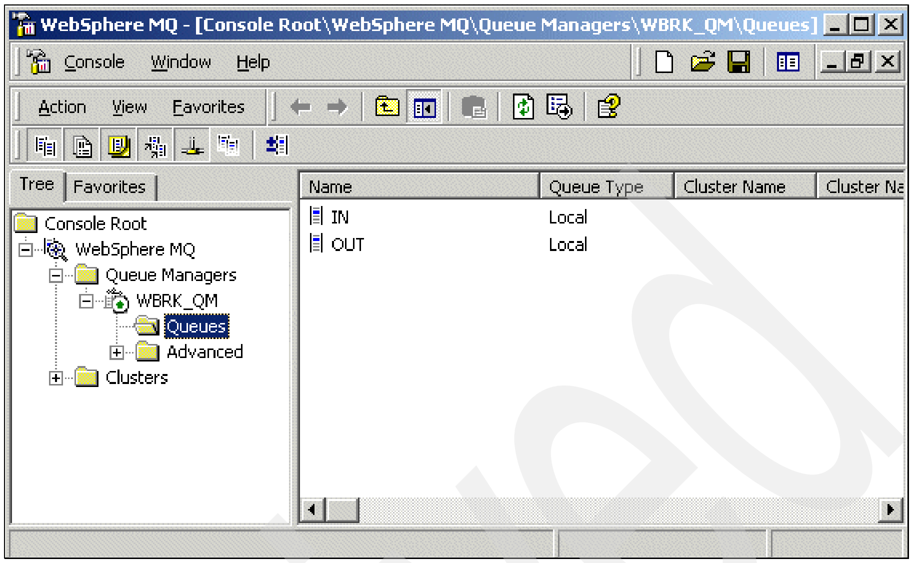
You cannot administer WebSphere MQ objects from within the Message Brokers Toolkit, so use the the WebSphere MQ Explorer snap-in to create the WebSphere MQ local queues.

To open the the WebSphere MQ Explorer snap-in, click **Start** → **Programs** → **IBM WebSphere MQ** → **WebSphere MQ Explorer**.

To create the IN and OUT local queues:

1. In WebSphere MQ Explorer, in the navigation tree, expand **WBRK_QM**, which is the name of the queue manager that hosts the broker.
2. Right-click the **Queues** folder under the WBRK_QM queue manager, then click **New** → **Local Queue** to open the Create Local Queue dialog box.
3. In the Create Local Queue dialog box, in the Queue Name field, type `IN`. Click **OK** to create a local queue called IN.
4. Repeat Step 3 to create a local queue called OUT.

   WebSphere MQ Explorer now displays two queues, called IN and OUT, that belong to the WBRK_QM queue manager, as shown in Figure 4-3 on page 89.



| Name | Queue Type | Cluster Name | Cluster Na |
|---|---|---|---|
| IN | Local | | |
| OUT | Local | | |

Figure 4-4 The local queues that are used by the BookstoreFlow message flow

The next section describes how to configure the nodes to connect to the correct queues.

## Configuring the node properties

To configure the properties of a node, start by opening the Properties dialog box:

1. Click the **Selection** button at the top of the Node palette.
2. Right-click the node, then click **Properties**.

### *Configuring the MQInput node*

To configure the MQInput node:

1. Open the Properties dialog box for the MQInput node.
2. On the Basic page in the Queue Name field, type `IN`, then click **OK** to close the Properties dialog box.
3. Save BookstoreFlow.msgflow. The error on the MQInput node is no longer displayed.

### *Configuring the MQOutput node*

To configure the MQOutput node:

1. Open the Properties dialog box for the MQOutput node.

2. On the Basic page in the Queue Name field, type `OUT`. Leave the Queue Manager Name field empty because the OUT queue is on the same queue manager as the IN queue. Click **OK** to close the Properties dialog box.

These properties are not mandatory in an MQOutput node. It is possible, instead, to set the information about the output queue within the header of messages that are processed by the message flow. However, for the BookstoreFlow message flow, type the information in the Properties dialog box.

#### *Configuring the Compute node*

Do not edit any of the properties in the Compute node, but look at the properties to see what has caused the error:

1. Open the Properties dialog box for the Compute node.
2. On the Basic page of the Properties dialog box, notice that the ESQL Module field contains `BookstoreFlow_Compute`. This value is entered automatically when the node is created because the Compute node must contain some ESQL. ESQL is held in a separate file called, in this case, BookstoreFlow.esql. However, an error is displayed in the Tasks view because the file BookstoreFlow.esql does not yet exist.
3. Click **Cancel** to close the Properties dialog box without saving any changes.

The next section describes how to create BookstoreFlow.esql and how to write some simple ESQL for the Compute node.

## 4.2.3 Writing ESQL for the Compute node

All of the ESQL that belongs to a message flow is stored, by default, in a single file. In this case, all of the ESQL for the BookstoreFlow message flow is stored in a file called BookstoreFlow.esql.

### Creating the ESQL file

To create BookstoreFlow.esql, right-click the **Compute** node, then click **Open ESQL**.

BookstoreFlow.esql does not already exist, so the Message Brokers Toolkit creates the file in the BookstoreFlow Messageflow Project. When the BookstoreFlow.esql file is created, it automatically opens in the ESQL editor.

When the BookstoreFlow.esql file is created, it already contains the minimum ESQL that is needed for the Compute node to successfully validate. If the ESQL file does not automatically generate some ESQL, do the following:

1. Close both BookstoreFlow.esql and BookstoreFlow.msgflow.

2. Open **BookstoreFlow.msgflow**, then right-click the Compute node and click **Open ESQL**.

Save BookstoreFlow.esql, then click the **BookstoreFlow.msgflow** tab to return to the Message Flow Editor. The error on the Compute node is no longer displayed and no items relating to this task appear in the Tasks view.

Click the **BookstoreFlow.esql** tab to display BookstoreFlow.esql in the ESQL editor again.

## The BookstoreFlow_Compute ESQL module

In the BookstoreFlow.esql file, there is a single module of ESQL called BookstoreFlow_Compute. This is the ESQL module that is referenced from the Compute node Properties dialog box. If you add another Compute node, or a Database node, to the BookstoreFlow message flow, a new ESQL module is created in BookstoreFlow.esql for the new node (Example 4-2).

*Example 4-2 ESQL for the BookstoreFlow_Compute module*

```
CREATE COMPUTE MODULE BookstoreFlow_Compute
   CREATE FUNCTION Main() RETURNS BOOLEAN
   BEGIN
      --CALL CopyMessageHeaders();
      --CALL CopyEntireMessage();
      RETURN TRUE;
   END;

   CREATE PROCEDURE CopyMessageHeaders() BEGIN
      DECLARE I INTEGER 1;
      DECLARE J INTEGER CARDINALITY(InputRoot.*[]);
      WHILE I < J DO
         SET OutputRoot.*[I] = InputRoot.*[I];
         SET I = I + 1;
      END WHILE;
   END;

   CREATE PROCEDURE CopyEntireMessage() BEGIN
      SET OutputRoot = InputRoot;
   END;
END MODULE;
```

The ESQL that is generated automatically does not produce any output. When the Compute node receives the input message, it builds an output message without any content. Therefore, you must edit the ESQL:

1. The fourth line (`--CALL CopyMessageHeaders();`) and the fifth line (`--CALL CopyEntireMessage();`) of the BookstoreFlow_Compute module are

commented out so that the Compute module does not parse them. To uncomment the lines, delete `--` from the start of each line.

2. Save BookstoreFlow.esql.

The Compute node now parses the lines that instruct the Compute node to copy to the output message the header and content of the input message. The message that the MQOutput node puts to the OUT queue has the same content as the message that the MQInput node got from the IN queue.

The next section describes how to test the BookstoreFlow message flow.

## 4.2.4 Testing the BookstoreFlow message flow

To test the BookstoreFlow message flow, you must deploy it to the broker. The broker then processes a test input message using the deployed message flow. When you perform administration tasks such as deploying a message flow application, work in the Broker Administration perspective.

### Deploying the BookstoreFlow message flow

While you are developing the message flow, you need to be able to test it quickly and easily on a non-production system. Perform a full deployment (as described in Chapter 5, "Administration" on page 133) to a broker on a production system only when you are sure that the message flow is complete.

To test the BookstoreFlow message flow, use the Rapid Application Development (RAD) method to deploy it to the broker called WBRK_BROKER. RAD enables you to create a configuration that you can use to repeatedly deploy the BookstoreFlow message flow to the broker. Using RAD, you initially define an execution group on the broker and publish the message flow to that execution group. When you subsequently make changes to the message flow, reuse this configuration to quickly deploy the changes to the broker.

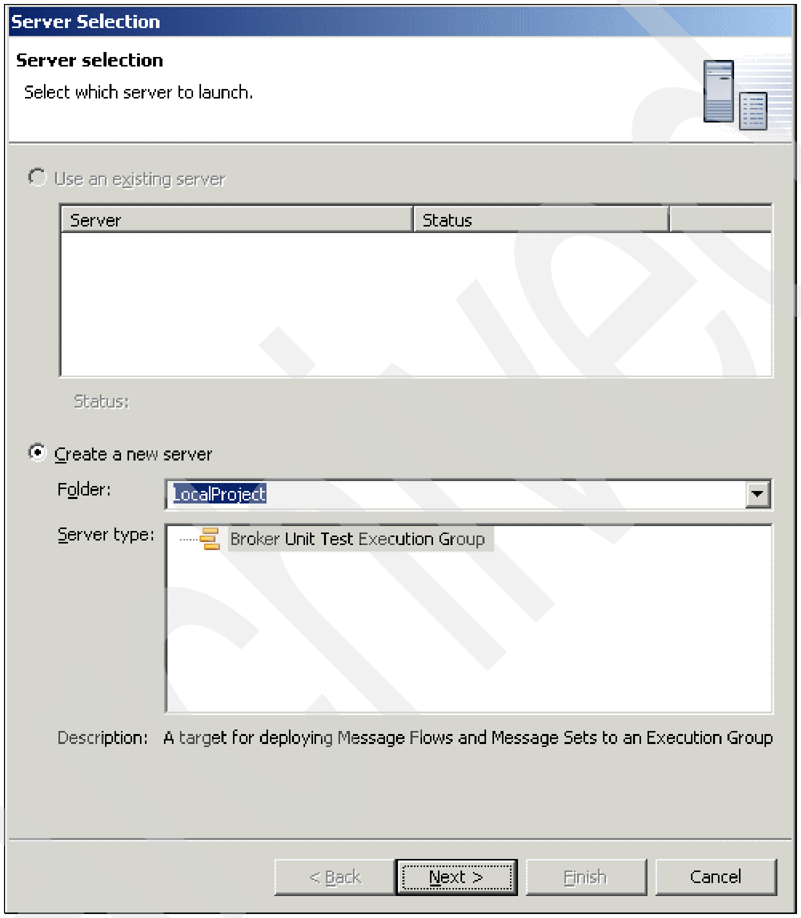
To prepare to deploy a message flow using RAD:

1. Create a new server.
2. Connect to the Configuration Manager.
3. Create an execution group.
4. Select the message flow project to deploy.

To deploy the BookstoreFlow message flow using RAD:

1. Switch to the Broker Administration perspective (click **Window** → **Open Perspective** → **Broker Administration**).

2. In the Broker Administration Navigator view, expand **Message Flows**, then right-click **Bookstore Messageflow Project** and click **Run on Server**. The Server Selection wizard opens.
3. In the Server Selection wizard (Figure 4-5), click **Create a new server** then click **Next**.


Server Selection

Server selection

Select which server to launch.

( ) Use an existing server

| Server | Status | |
|---|---|---|

Status:

(●) Create a new server

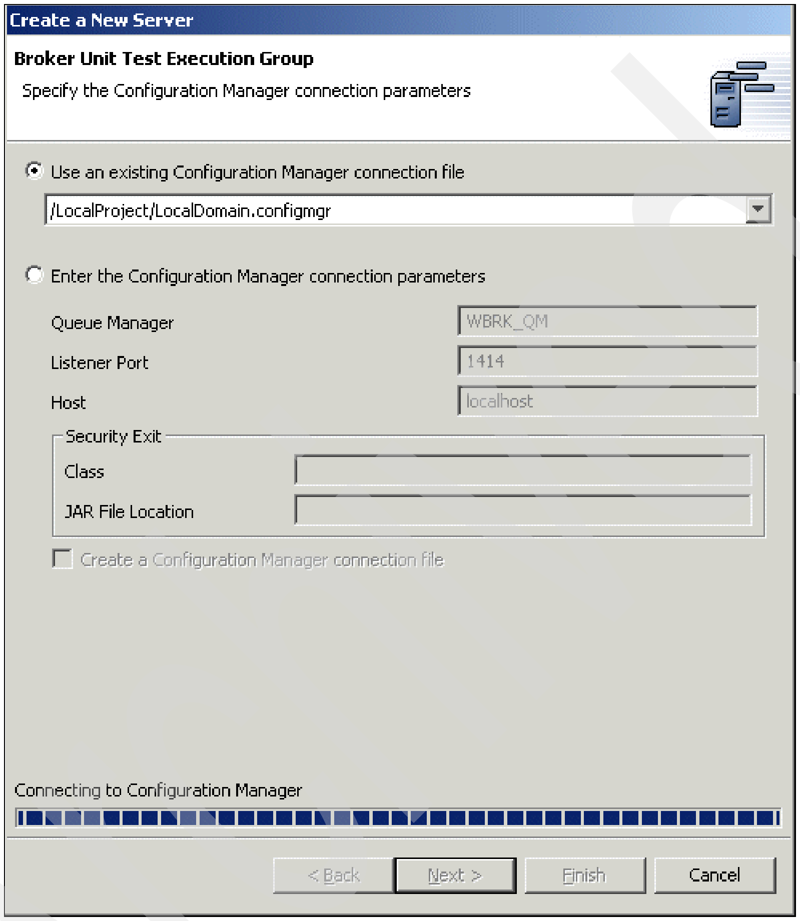
Folder: LocalProject

Server type: Broker Unit Test Execution Group

Description: A target for deploying Message Flows and Message Sets to an Execution Group

Figure 4-5 Configuring RAD: Creating a new server

4. Ensure that **Use an existing Configuration Manager connection file** is selected (Figure 4-6), then click **Next**. Wait while the wizard connects to the Configuration Manager.


Create a New Server
Broker Unit Test Execution Group
Specify the Configuration Manager connection parameters

(● selected) Use an existing Configuration Manager connection file

/LocalProject/LocalDomain.configmgr

Enter the Configuration Manager connection parameters

Queue Manager: WBRK_QM

Listener Port: 1414

Host: localhost

Security Exit

Class:

JAR File Location:

☐ Create a Configuration Manager connection file

Connecting to Configuration Manager

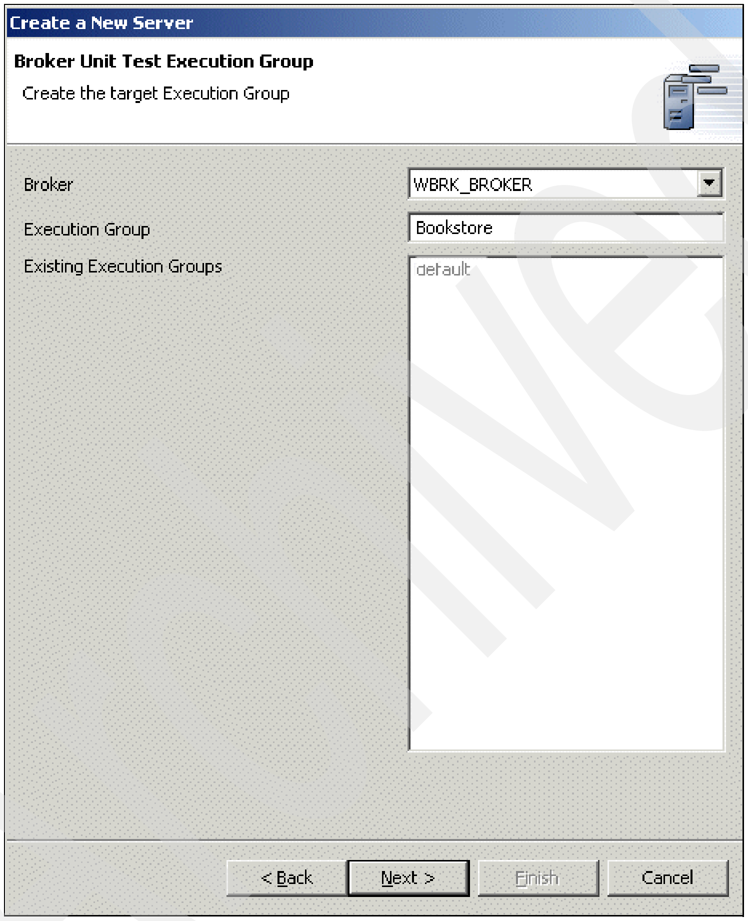
< Back
Next >
Finish
Cancel

Figure 4-6 Configuring RAD: Connecting to the Configuration Manager

5. Create the execution group to contain the message flow application on the broker (Figure 4-7):

   a. Ensure that the Broker field contains `WBRK_BROKER`.

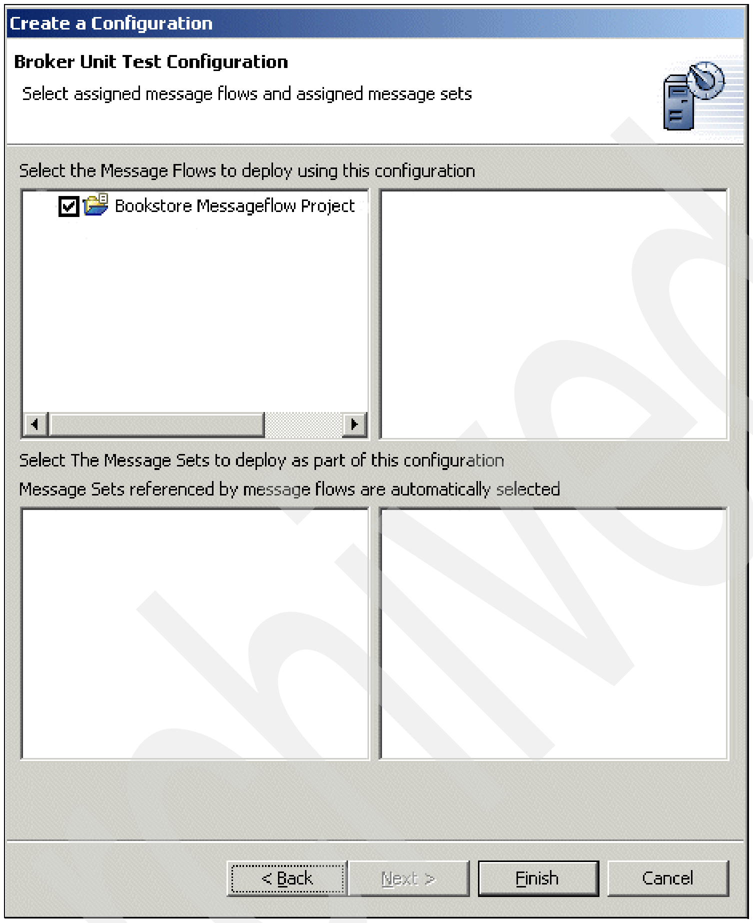
   b. In the Execution Group field, type `Bookstore` to create an execution group called Bookstore, then click **Next**.


Create a New Server
Broker Unit Test Execution Group
Create the target Execution Group
< Back
Next >
Finish
Cancel

Broker: WBRK_BROKER
Execution Group: Bookstore
Existing Execution Groups: default

*Figure 4-7 Configuring RAD: Creating a new execution group*

6. Select the BookstoreFlow message flow (Figure 4-8) and click **Finish**.



Create a Configuration

Broker Unit Test Configuration

Select assigned message flows and assigned message sets

Select the Message Flows to deploy using this configuration

☑ Bookstore Messageflow Project

Select The Message Sets to deploy as part of this configuration

Message Sets referenced by message flows are automatically selected

*Figure 4-8 Selecting the BookstoreFlow message flow to deploy to the broker*

The BookstoreFlow message flow is deployed to the BookstoreFlow execution group on the broker.

When the message flow is successfully deployed, you might see this error message:

```
Cannot start the server because it is already running.
```

You can safely ignore this message.

## Running BookstoreFlow with an XML test message

To test the BookstoreFlow message flow, use the Enqueue editor in the Message Brokers Toolkit to put an XML input message on the IN queue. The BookstoreFlow message flow gets the input message from the IN queue, processes the message, and then puts an output message on the OUT queue. The Enqueue editor enables you to easily create an XML message without having to understand how to create message headers. The Enqueue editor automatically adds an MQMD header to the message.

To create an enqueue file to which you add the message content:

1. On the toolbar, click the small arrow on the Enqueue button (second from left in Figure 4-9), then click **Put Message**. The New Enqueue Message File wizard opens.

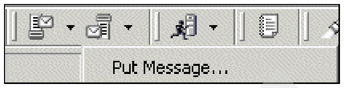
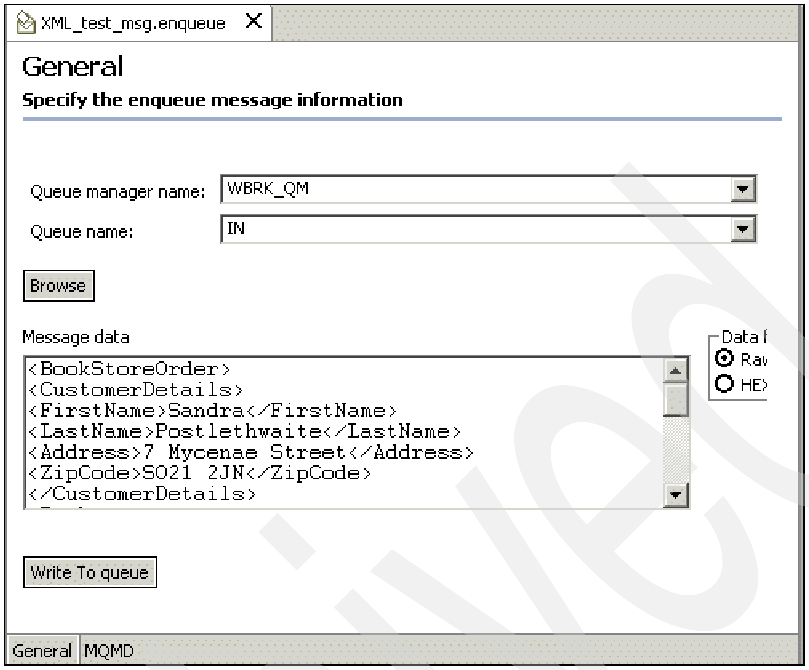


*Figure 4-9 The Dequeue and Enqueue buttons on the toolbar*

2. Click **BookstoreFlow Messageflow Project**, then in the File Name field type `Bookstore_msg`. The new enqueue file, XML_test_msg.enqueue, opens in the Enqueue editor (Figure 4-10 on page 100).


XML_test_msg.enqueue

General

Specify the enqueue message information

Queue manager name: WBRK_QM

Queue name: IN

Message data

```
<BookStoreOrder>
<CustomerDetails>
<FirstName>Sandra</FirstName>
<LastName>Postlethwaite</LastName>
<Address>7 Mycenae Street</Address>
<ZipCode>SO21 2JN</ZipCode>
</CustomerDetails>
```

Data f

[✓ Raw] HEX

*Figure 4-10 The Enqueue editor*

3. In the Queue manager name field, type `WBRK_QM`.
4. In the Queue name field, type `IN`.
5. In the Message data field, type the XML message content shown in Example 4-1 on page 84 (at the beginning of this chapter).
6. Save XML_test_msg.enqueue.
7. Click the **MQMD** tab at the bottom of the Enqueue editor to display the default MQMD message header settings. Do not edit these settings. The Enqueue editor builds an XML input message using the MQMD message header settings and the text that you typed in the Message data field.
8. Click the **General** tab, then click the **Write To queue** button to put the XML input message on the IN queue. A message is displayed to confirm that the message was successfully put to the queue.

To get the output message from the OUT queue:

1. Click the **Dequeue** button on the toolbar (see Figure 4-9 on page 99) to open the Dequeue Message wizard.
2. In the Queue Manager Name field, type `WBRK_QM`, and in the Queue Name field type `OUT`. The Dequeue Message wizard is case-sensitive.

3. Click **Read From Queue**. The content of the output message is displayed in the wizard.

If you receive the following Dequeue error message, the BookstoreFlow message flow has not put an output message on to the OUT queue:

```
BIP0917E No messages found to dequeue
```

To diagnose the problem:

1. Open the WebSphere MQ Explorer snap-in and look at the Current Depth column for the IN and OUT queues. If there is a zero (0) in the Current Depth column for the OUT queue and a one (1) in the Current Depth column for the IN queue, the input message has either never left the IN queue, or BookstoreFlow message flow encountered a problem during processing and rolled back the input message to the IN queue.
1. Use the Dequeue Message wizard to get the message from the IN queue so that the Current Depth of the IN queue is zero. In the WebSphere MQ Explorer snap-in, click the **Refresh** button on the toolbar to refresh the view of the queues.
2. Open the WebSphere MQ Services snap-in and verify that the following objects are running:
   - WBRK_QM queue manager
   - WBRK_QM listener
   - WBRK_QM channel initiator
   - WBRK_QM command server

   If any of these objects have stopped, restart them by right-clicking the object, then clicking **All tasks** → **Start**.
3. In the BookstoreFlow message flow, make sure that the queue names in the MQInput and MQOutput nodes are spelled correctly and are in upper case.
4. If none of the previous suggestions solve the problem, see the Problem Determination chapter for a more comprehensive list of things to check.

## 4.3 Developing a message set

The BookstoreFlow message flow processes a simple XML message, the structure of which is defined in the message itself. There is no external definition against which the BookstoreFlow message flow can verify the message structure or refer to it for information about the fields included in the message. Therefore, the BookstoreFlow message flow is limited in the amount of manipulation that it can perform on the message. To program the message flow to perform more complex functions, predefine the message structure in an external message set.

A message set is a template that defines the structure of the messages that are processed by a message flow. A message definition is held externally to the message in the message set, and, when the message set is deployed, the definition is compiled into a dictionary. When the message flow is processing a message, the message flow can refer to this dictionary that is held in the broker.

When you program message flows, you do not have to define the messages in a message set; you can use self-defining XML messages effectively, depending on type of processing that you need the message flow application to perform. You might decide, for example, that to process a simple message, it is not worth the extra effort of predefining the message structure in a message set, which can be complex and time-consuming. However, if you do predefine the messages in a message set, it is subsequently much easier to program a message flow to perform complex manipulations of messages. This is partly because there are several built-in nodes that can do some of the programming for you. For example, if you use a Mapping node, you can map fields from one message to another without having to write the complex ESQL that you would have to write for a Compute node to perform the equivalent function.

Example 4-3 gives an example of some ESQL that you could write to create an output message called ShippingDetails from the BookStoreOrder input message. The first part of the ESQL copies the message headers from the input message to the new output message. The collection of SET statements in the second part of the ESQL performs the actual mapping of input message fields onto output message fields.

**Attention:** Note that the comment lines and the SET statements are wrapped across lines. If you insert the ESQL in a message flow, do not duplicate the line breaks for these lines.

*Example 4-3 ESQL that could perform the equivalent function to the Mapping node in the modified version of the BookstoreFlow message flow*

```
CREATE COMPUTE MODULE BookstoreFlow_Compute
   CREATE FUNCTION Main() RETURNS BOOLEAN

--The following ESQL copies the header from the input message
--to the output message
   BEGIN
   DECLARE I INTEGER;
   SET I=1;
   WHILE I<CARDINALITY(InputRoot.*[]) DO
      SET OutputRoot.*[I] = InputRoot.*[I];
      SET I=I+1;
   END WHILE;

--The following ESQL maps some of the fields from the input message
--on to the relevant fields of the output message
   SET OutputRoot.XML.BookStoreOrder.CustomerDetails.FirstName =
InputRoot.XML.ShippingDetails.CustomerDetails.FirstName;
   SET OutputRoot.XML.BookStoreOrder.CustomerDetails.LastName =
InputRoot.XML.ShippingDetails.CustomerDetails.LastName;
   SET OutputRoot.XML.BookStoreOrder.CustomerDetails.Address =
InputRoot.XML.ShippingDetails.CustomerDetails.Address;
   SET OutputRoot.XML.BookStoreOrder.CustomerDetails.ZipCode =
InputRoot.XML.ShippingDetails.CustomerDetails.ZipCode;
   SET OutputRoot.XML.BookStoreOrder.Book.Title =
InputRoot.XML.ShippingDetails.BookDetails.Title;
   SET OutputRoot.XML.BookStoreOrder.Book.Author =
InputRoot.XML.ShippingDetails.BookDetails.Author;
   SET OutputRoot.XML.BookStoreOrder.BookOrder.Price =
InputRoot.XML.ShippingDetails.BookDetails.Price;

   RETURN TRUE;
   END;
END MODULE;
```

The ESQL in Example 4-3 on page 102 is relatively short as the manipulation that the ESQL is performing is relatively simple. However, for more complex manipulations, the amount and complexity of the ESQL would increase. When you have created the message set that is described in the following sections, compare the ease of writing the ESQL in Example 4-3 on page 102 with mappings that you create in the Mapping editor using the predefined message structures.

The following sections look at how to modify the BookstoreFlow message flow to generate a message that confirms that the customer's order has been shipped to their address.

## 4.3.1 Modeling the message structure

The message model is the structure of the message that is defined in the message definition file. You can create one or more message definition files to represent different formats of the message model, such as XML, Tagged Delimited String (TDS), or Custom Wire Format (CWF). For more information about the different message formats that are supported in WebSphere Business Integration Message Broker, see the product documentation: **Concepts** → **Message modelling** → **Physical formats in the MRM domain**.

The Bookstore Messageset contains one message definition for XML messages. The message definition, *Bookstore*, defines two message models:

- BookStoreOrder, the message that is sent when the customer orders a book from the online bookstore.
- ShippingDetails, the message that is generated by the BookstoreFlow message flow to confirm that the order has been despatched to the customer.

## Creating the message set

Before you can start modelling the message structure, create the message set that holds the message definition:

1. Create Bookstore Messageset Project and Bookstore Messageset:
   a. Click **File** → **New** → **Message Set Project**.
   b. In the Project Name field, type `Bookstore Messageset Project`, then click **Next**.
   c. In the Message Set Name field, type `Bookstore Messageset`, then click **Next**.
   d. Select **XML Wire Format Name** and keep the default name `XML1`. Leave the other check boxes empty. This setting defines the physical format of the message as being XML. Click **Finish**.

   A new project called Bookstore Messageset Project is displayed in the Resource Navigator view. A message set called Bookstore Messageset is displayed in Bookstore Messageset Project. The messageSet.msgset file opens automatically in the Message Set Editor.
2. Create the Bookstore message definition file in the Bookstore Messageset Project:
   a. In the Resource Navigator view, click **Bookstore Messageset Project** to highlight it.
   b. Click **File** → **New** → **Message Definition File**.
   c. Click **Create a new message definition file**, then click **Next**.
   d. Click **Bookstore Messageset** to highlight it, then in the File Name field, type `Bookstore`. Click **Finish**.

A new message definition file called Bookstore.mxsd is displayed in Bookstore Messageset in the Resource Navigator view. The Bookstore.mxsd file opens automatically in the Message Definition editor. The Resource Navigator view should look similar to Figure 4-11 on page 105.

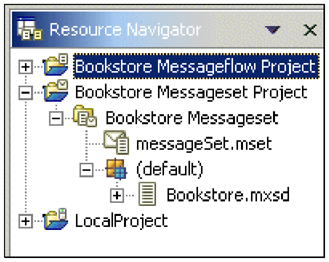


*Figure 4-11 The message set resources*

Bookstore.mxsd is displayed in the Default namespace of the Bookstore Messageset Project. For more information about namespaces, see the product documentation: **Concepts** → **Message modelling** → **The message model** → **Namespaces**.

## Configuring the message set

In case the BookstoreFlow message flow cannot deduce the format of the message (for example, XML) from the message header, the message flow uses the default setting that is defined in the message set. This default format is set in the message set properties file, messageSet.msgset.

To configure the default format of the message set:

1. Open **messageSet.msgset** in the Message Set editor.
2. In the Properties Hierarchy, click **Message Set**.
3. From the Default Wire Format menu, select **XML1**, then save the file.

The term *wire format* is being deprecated, so this book refers to the *physical format* or *message format* but the terms are synonymous. The physical format of the message is the way in which the message data is physically organized for sending. The logical format, by comparison, is the logical organization of the message content: the structure of the message.

For more information about physical message formats, see the product documentation at **Concepts** → **Message modelling** → **Physical formats in the MRM domain**. For more information about logical message model, go to **Concepts** → **Message modelling** → **The message model**.

For a useful example of how to model the same message in different physical formats and transform the message from one physical format to another, explore the Video Rental sample in the Samples Gallery of the product documentation.

## Defining the logical structure of the BookStoreOrder message

When you created the Bookstore Messageset, you defined the physical format of the message, BookStoreOrder, as XML. Now define the logical structure of the content within the message.

The BookStoreOrder message shown at the beginning of this chapter in Example 4-1 on page 84 is constructed from several elements of various types. For information about defining more complex message structures, see the product documentation at **Concepts** → **Message modelling** → **The message model**.

The logical message structure of the BookStoreOrder message is constructed from 18 elements (Figure 4-12). Each element represents a field in the message.

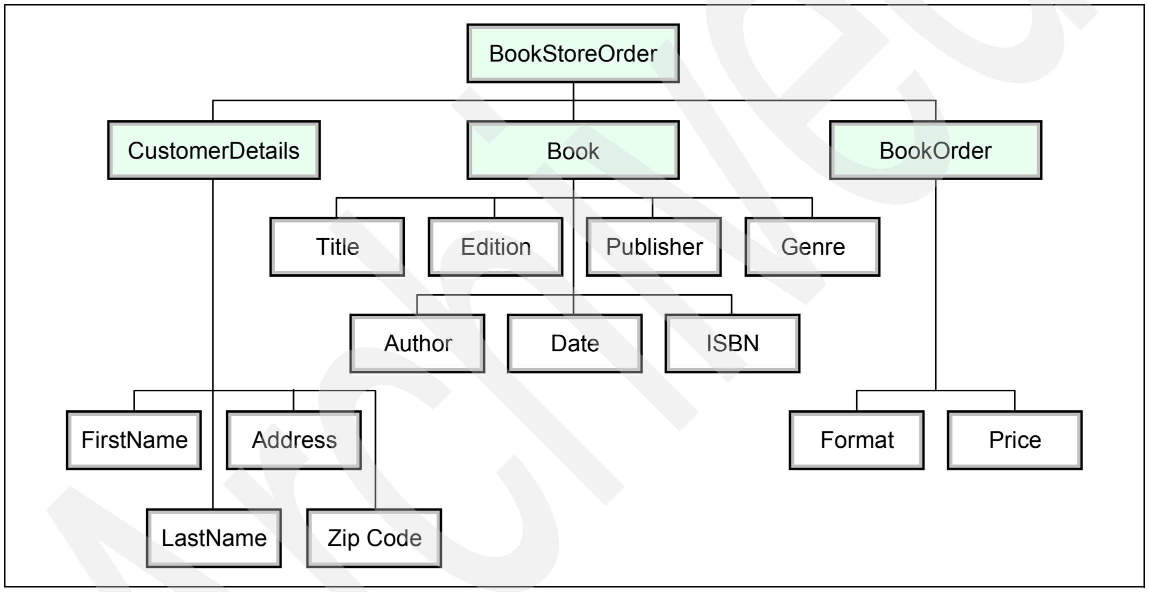


*Figure 4-12 Logical structure of BookStoreOrder message—shaded elements based on complex types*

Thirteen of the elements, or fields, in the BookStoreOrder message directly contain data, for example, FirstName contains the data `Sandra`, and Title contains the data `Sitting Pretty` (refer to Example 4-1 on page 84). Each of these elements is based on simple types of data and are shown unshaded in Figure 4-12. For example, the FirstName element is based on the simple type *string*, and the Edition element is based on the simple type *int*.

The remaining five elements, which are shaded in the figure, are based on complex types. Complex types define the structure of a message. Elements represent the fields of a message but complex types define the order in which the fields occur in the message.

The CustomerDetails element is based on the CustomerDetails complex type, which is constructed from four elements: FirstName, LastName, Address, and ZipCode. So within the CustomerDetails part of the message, there are fields for customer's first name, last name, address, and zip code.

To define the logical message structure of the BookStoreOrder message:

1. Open the **Bookstore.mxsd** file in the Message Definition editor.
2. Add the message element that contains all other elements in the message:
   a. In Bookstore.mxsd, right-click **Messages**, then click **Add Message** (Figure 4-13).

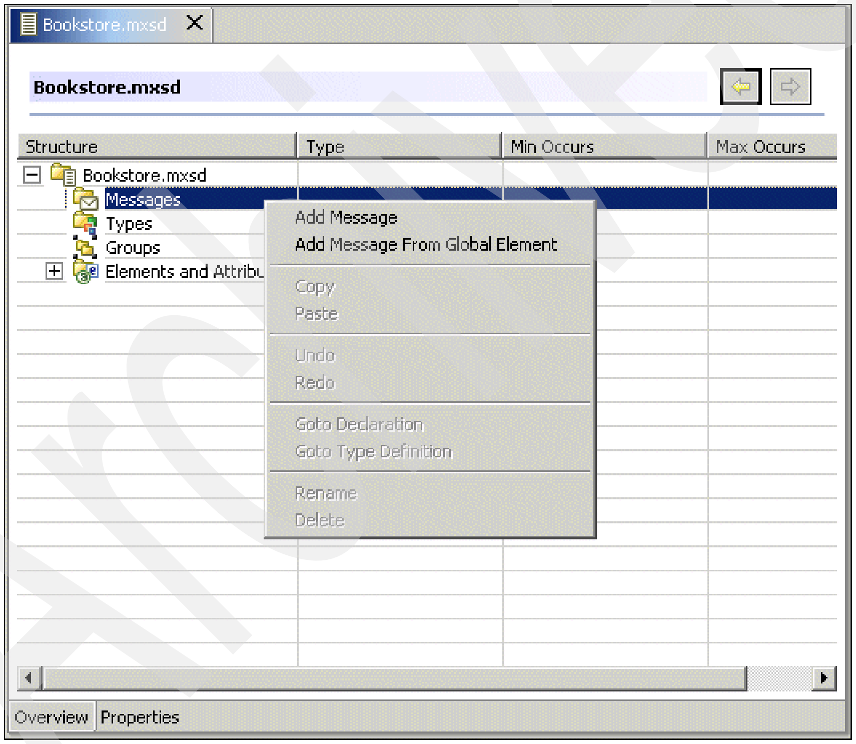


*Figure 4-13 Adding the message element to define the root of the message*

   b. Type `BookStoreOrder`, which is the name of the message, then press Enter.

3. Add the elements to represent the fields of the message:

   a. In Bookstore.mxsd, right-click **Elements and Attributes**, then click **Add Global Element** (Figure 4-14).

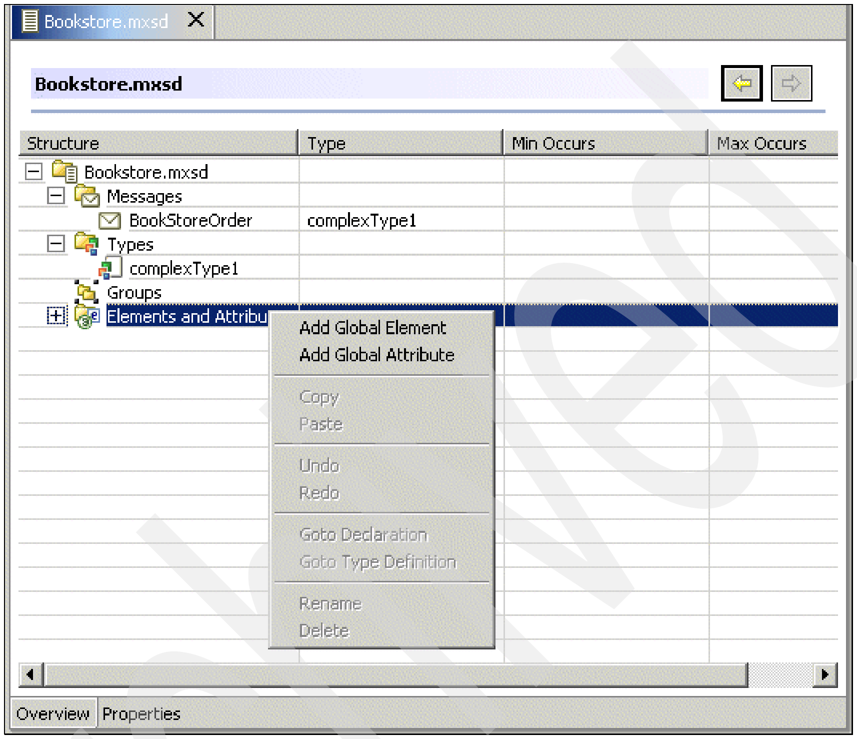


*Figure 4-14   Adding a global element to the message definition*

   b. Type the name of the element (for example, `FirstName`), then press Enter.

   c. Repeat steps 3a and 3b for each of the elements in the first column of Table 4-1. Ignore the second column for now.

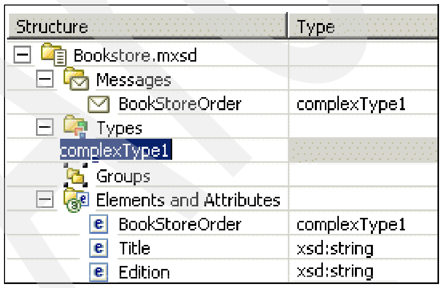
*Table 4-1   The elements in the BookStoreOrder message*

| **Elements** | **Type** |
|---|---|
| Title | xsd:string |
| Edition | xsd:int |
| Author | xsd: |
| Date | xsd:date |
| Genre | xsd:string |

| Elements | Type |
|---|---|
| ISBN | xsd:string |
| Publisher | xsd:string |
| Format | xsd:string |
| Price | xsd:float |
| FirstName | xsd:string |
| LastName | xsd:string |
| Address | xsd:string |
| ZipCode | xsd:string |
| CustomerDetails | CustomerDetails (a complex type) |
| Book | Book (a complex type) |
| BookOrder | BookOrder (a complex type) |
| BookStoreOrder | BookStoreOrder (a complex type) |

4. For each of the elements that is based on a complex type (the elements that are shaded in Figure 4-12 on page 106), define the complex type. Create the complex types:

   a. The first complex type was created automatically when you created the BookStoreOrder message element. The default name of the complex type is complexType1. Click **complexType1** to select it, then click it again and type over it with the name `BookStoreOrder`, as shown in Figure 4-15.



| Structure | Type |
|---|---|
| Bookstore.mxsd | |
| Messages | |
| BookStoreOrder | complexType1 |
| Types | |
| complexType1 | |
| Groups | |
| Elements and Attributes | |
| BookStoreOrder | complexType1 |
| Title | xsd:string |
| Edition | xsd:string |

*Figure 4-15 Renaming a complex type*

b. Right-click **Types**, then click **Add Complex Type** (Figure 4-16).

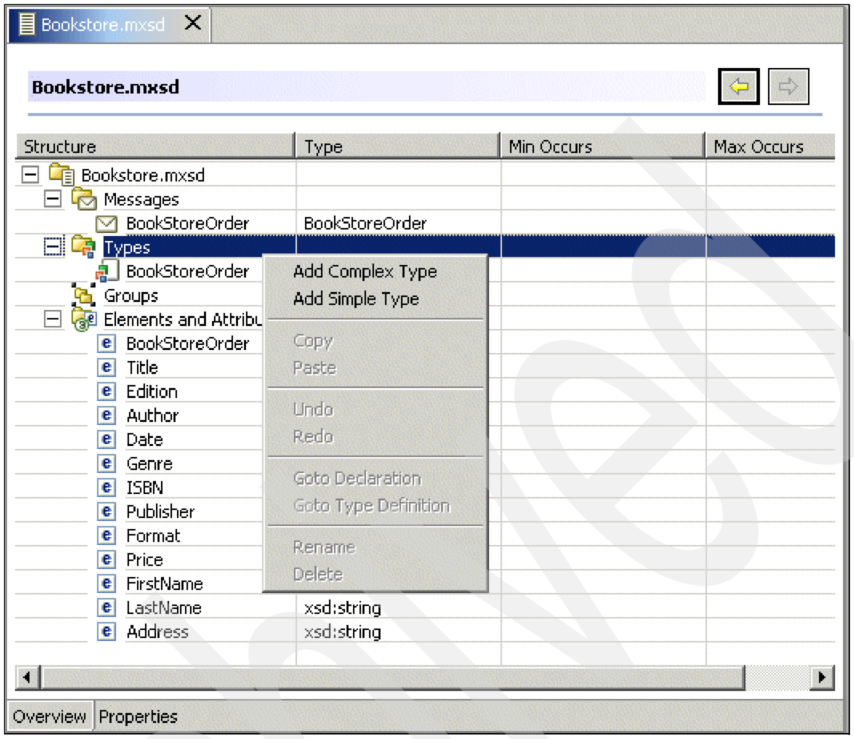


*Figure 4-16 Creating a complex type*

c. Type the name of the complex type, for example, `CustomerDetails`, then press Enter.

d. Repeat steps 4 on page 109a on page 109 and 4 on page 109b to create complex types for each of the complex elements listed in Table 4-1 on page 108.

5. Define the structure of the complex types by adding references to the elements on which they are based:

   a. In Bookstore.mxsd, right-click the name of the complex type (for example, `CustomerDetails`), then click **Add Element Reference** (Figure 4-17).

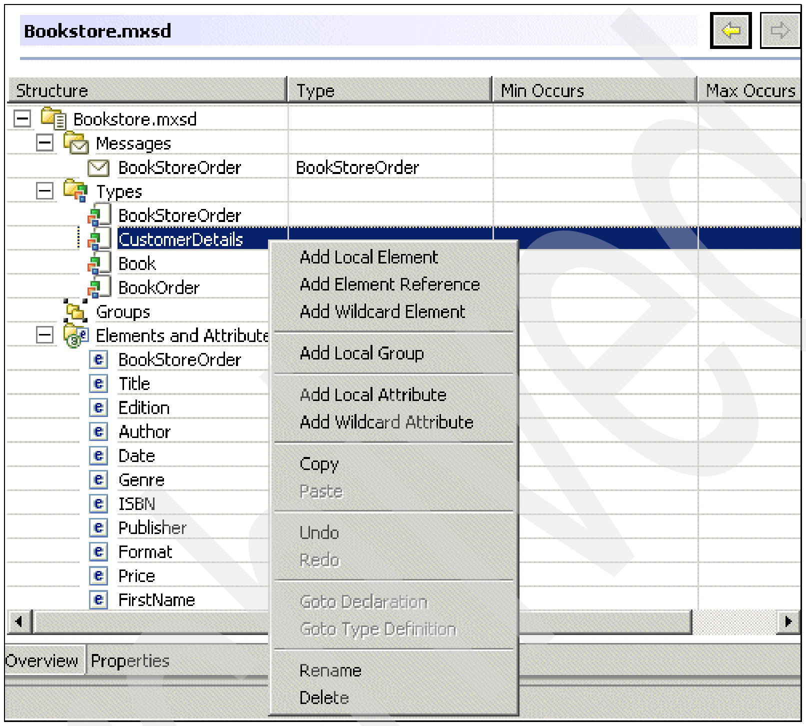
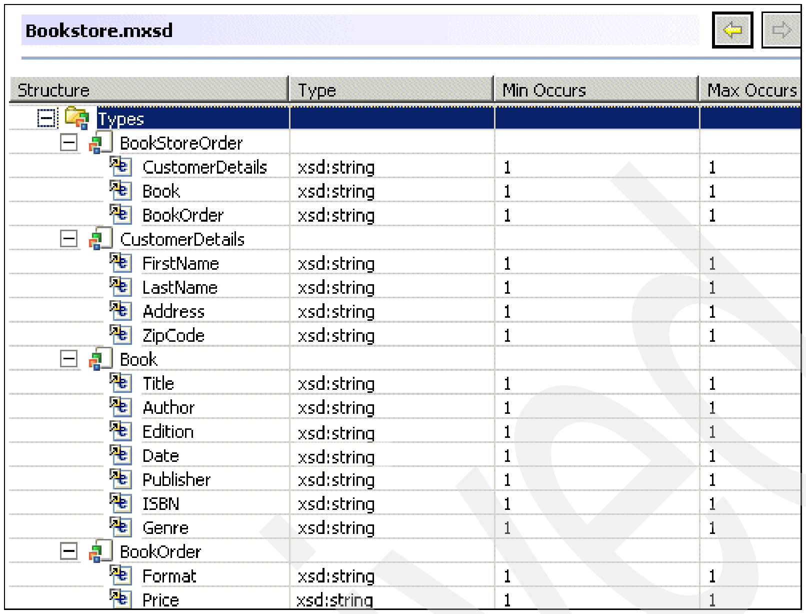


*Figure 4-17 Add element references to define complex types structure*

   b. From the Element Reference list, select the name of the element that is first in structure. For example, for the CustomerDetails complex type, select **FirstName**.

   c. Repeat steps 5a and 5b to add all of the element references to each complex type. Refer to Example 4-3 on page 102 for the names of the elements. The structure of the complex types should look like Figure 4-18 on page 112.



Bookstore.mxsd

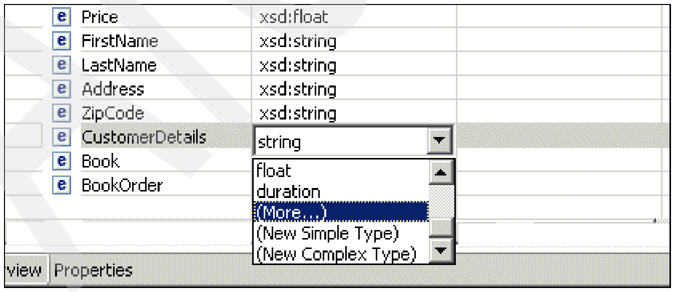
| Structure | Type | Min Occurs | Max Occurs |
|---|---|---|---|
| Types | | | |
| BookStoreOrder | | | |
| CustomerDetails | xsd:string | 1 | 1 |
| Book | xsd:string | 1 | 1 |
| BookOrder | xsd:string | 1 | 1 |
| CustomerDetails | | | |
| FirstName | xsd:string | 1 | 1 |
| LastName | xsd:string | 1 | 1 |
| Address | xsd:string | 1 | 1 |
| ZipCode | xsd:string | 1 | 1 |
| Book | | | |
| Title | xsd:string | 1 | 1 |
| Author | xsd:string | 1 | 1 |
| Edition | xsd:string | 1 | 1 |
| Date | xsd:string | 1 | 1 |
| Publisher | xsd:string | 1 | 1 |
| ISBN | xsd:string | 1 | 1 |
| Genre | xsd:string | 1 | 1 |
| BookOrder | | | |
| Format | xsd:string | 1 | 1 |
| Price | xsd:string | 1 | 1 |

*Figure 4-18 Constructing the complex types*

6. Now that you have defined the structure of the complex types, specify the type of each element so that the complex type structure is imposed on the elements:

   a. In the Elements and Attributes section, click an element to highlight it (for example, Edition).

   b. Click the **Type** column next to the element's name to display a pull-down menu (Figure 4-19).

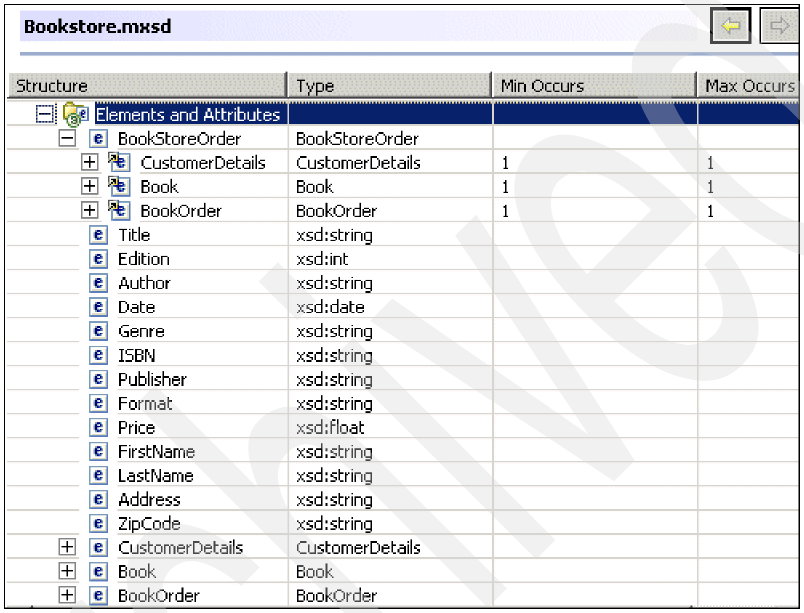


| | |
|---|---|
| Price | xsd:float |
| FirstName | xsd:string |
| LastName | xsd:string |
| Address | xsd:string |
| ZipCode | xsd:string |
| CustomerDetails | string |
| Book | |
| BookOrder | |

*Figure 4-19 Setting the types of the elements*

   c. From the list, select the type (for example, xsd:int or BookOrder), then press Enter. If the type is not available in the list, scroll down the menu and click **(More...)** to open a dialog box listing more simple types and all of the complex types that you created in step 3 on page 108.

d. Repeat steps 6 on page 112a on page 112 through 6 on page 112c on page 112 for each of the elements in Table 4-1 on page 108. When you associate a complex type with an element (for example, CustomerDetails), the structure of elements beneath the CustomerDetails is automatically constructed according to the structure of the complex type (Figure 4-20).


Bookstore.mxsd

| Structure | Type | Min Occurs | Max Occurs |
|---|---|---|---|
| Elements and Attributes | | | |
| BookStoreOrder | BookStoreOrder | | |
| CustomerDetails | CustomerDetails | 1 | 1 |
| Book | Book | 1 | 1 |
| BookOrder | BookOrder | 1 | 1 |
| Title | xsd:string | | |
| Edition | xsd:int | | |
| Author | xsd:string | | |
| Date | xsd:date | | |
| Genre | xsd:string | | |
| ISBN | xsd:string | | |
| Publisher | xsd:string | | |
| Format | xsd:string | | |
| Price | xsd:float | | |
| FirstName | xsd:string | | |
| LastName | xsd:string | | |
| Address | xsd:string | | |
| ZipCode | xsd:string | | |
| CustomerDetails | CustomerDetails | | |
| Book | Book | | |
| BookOrder | BookOrder | | |

*Figure 4-20 Building the complex type structure*

7. In the Types section, expand each of the complex types and check that the structure of the message that you have modelled is the same as the structure of the BookStoreOrder message shown in Example 4-1 on page 84. If the order of the types is not correct, click and drag the types up or down the list until they are in the correct order.

## Defining the logical structure of the ShippingDetails message

In the Bookstore scenario, the message flow application generates a shipping confirmation message based on the BookStoreOrder message. The logical structure of the ShippingDetails message is shown in the diagram in Figure 4-19 on page 112.

To define the logical structure of the ShippingDetails message:

1. In Bookstore.mxsd, add a message called **ShippingDetails**.

2. When you add the message, a complex type called complexType1 is created automatically. In the Types section, rename complexType1 to `ShippingDetails`.
3. In the Elements and Attributes section, add a global element called **BookDetails**.
4. In the Types section, create a new complex type called **BookDetails**.
5. In the Elements and Attributes section, specify that the BookDetails element is based on the **BookDetails** complex type.
6. In the Types section, add to the ShippingDetails complex type, references to the **CustomerDetails** element and the **BookDetails** element so that the structure of the elements in the ShippingDetails message is similar to the diagram in Figure 4-21.

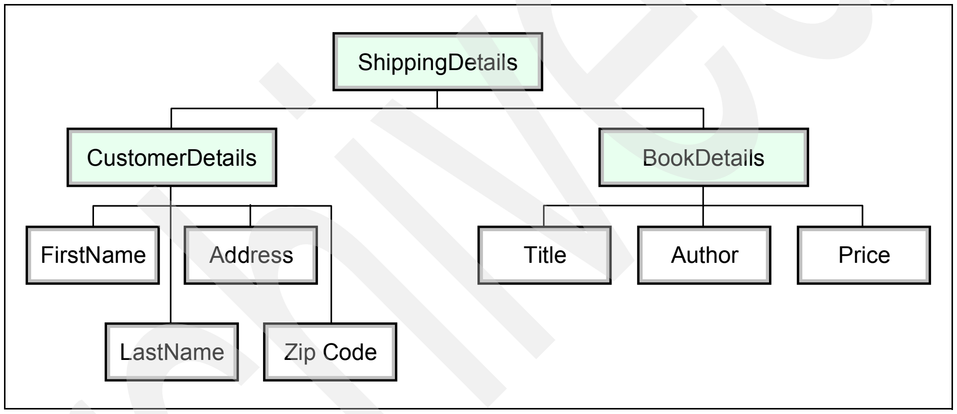
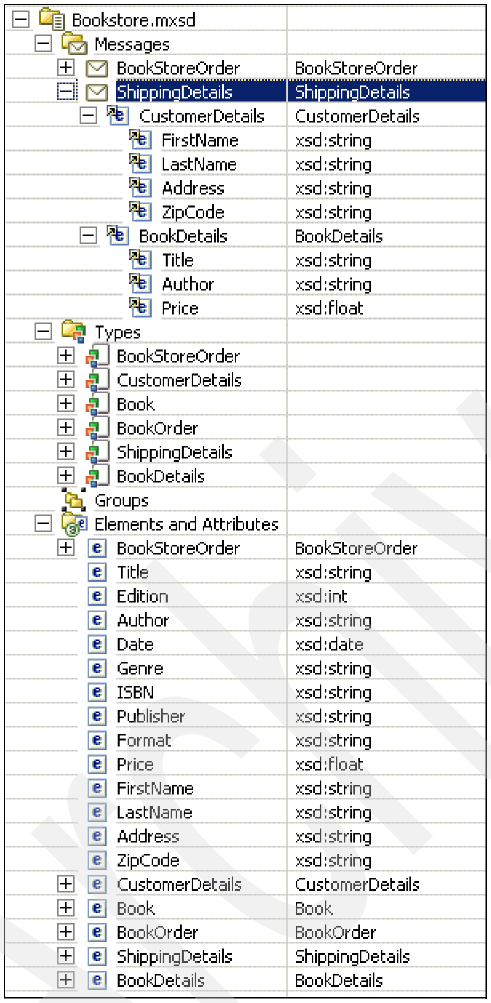


*Figure 4-21 Logical message structure of ShippingDetails message—shaded elements based on complex types*

Figure 4-22 shows the ShippingDetails message structure in the Message Definition editor.



| | |
|---|---|
| Bookstore.mxsd | |
| Messages | |
| BookStoreOrder | BookStoreOrder |
| ShippingDetails | ShippingDetails |
| CustomerDetails | CustomerDetails |
| FirstName | xsd:string |
| LastName | xsd:string |
| Address | xsd:string |
| ZipCode | xsd:string |
| BookDetails | BookDetails |
| Title | xsd:string |
| Author | xsd:string |
| Price | xsd:float |
| Types | |
| BookStoreOrder | |
| CustomerDetails | |
| Book | |
| BookOrder | |
| ShippingDetails | |
| BookDetails | |
| Groups | |
| Elements and Attributes | |
| BookStoreOrder | BookStoreOrder |
| Title | xsd:string |
| Edition | xsd:int |
| Author | xsd:string |
| Date | xsd:date |
| Genre | xsd:string |
| ISBN | xsd:string |
| Publisher | xsd:string |
| Format | xsd:string |
| Price | xsd:float |
| FirstName | xsd:string |
| LastName | xsd:string |
| Address | xsd:string |
| ZipCode | xsd:string |
| CustomerDetails | CustomerDetails |
| Book | Book |
| BookOrder | BookOrder |
| ShippingDetails | ShippingDetails |
| BookDetails | BookDetails |

*Figure 4-22 The ShippingDetails message modeled in the Message Definition editor*

You have defined the logical structure of both the BookStoreOrder message and the ShippingDetails message in Bookstore.mxsd. Next, modify the BookstoreFlow message flow so that it uses the Bookstore message set.

## 4.3.2 Modifying the BookstoreFlow message flow

The BookstoreFlow message flow currently contains three nodes:

- An MQInput node to retrieve the input message from the IN queue
- A Compute node to process the input message to generate an output message
- An MQOutput node to put the output message on the OUT queue.

The Compute node in the BookstoreFlow is capable of modifying the message and generating a new message. However, all information about the message structure is held in the message itself (the XML message is self-defining), so the Compute node must be programmed, using ESQL, to perform the processing on the message. The more complex the processing, the more complex the ESQL must be, which makes the message flow more complex to maintain and debug.

If the logical structure of the XML message is externally defined in a message set, you do not have to program such complex ESQL to manipulate the message. The message flow refers, instead, to the external message set for information about each message that it processes. If you create a message set, you can use a Mapping node instead of the Computer node.

The Mapping node enables you to load into the Mapping editor the input message (BookStoreOrder) and output message (ShippingDetails) structures from the message set. You can then simply drag and drop elements from the input message onto elements in the output message to show which data should be used from the input message to create the output message.

Mapping fields from one message to another is useful if, for example, the business application that sends the message to the broker structures the data in the message differently from the business application that will receive the message. If the first business application positions the customer’s first name before their last name but the second business application expects the customer’s last name to precede their first name in the message, the Mapping node can map the fields so that each business application can understand the message.

In the BookstoreFlow message flow, in addition to replacing the Compute node, edit the properties of the MQInput node so that it uses the correct message set to parse input messages. In addition, to avoid conflict between the original version of the BookstoreFlow message flow and the new version of BookstoreFlow message flow in the broker, before you deploy the new version of the BookstoreFlow message flow, you must create a new execution group and two new WebSphere MQ local queues.

## Replacing the Compute node with a Mapping node

To modify the BookstoreFlow message flow by replacing the Compute node with a Mapping node:

1. In the Resource Navigator view, double-click **BookstoreFlow.msgflow** to open the BookstoreFlow message flow in the Message Flow Editor.
2. In the Message Flow Editor, right-click the **Compute** node, then click **Delete**. The Compute node is deleted from the BookstoreFlow message flow.
3. From the node palette, select the **Mapping** node and add it to the canvas between the MQInput node and the MQOutput node.
4. Connect the nodes (Figure 4-23), then save BookstoreFlow.msgflow.

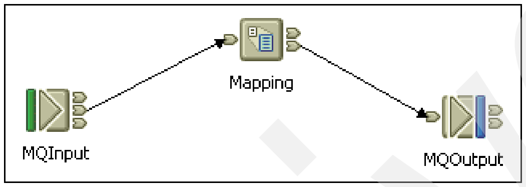
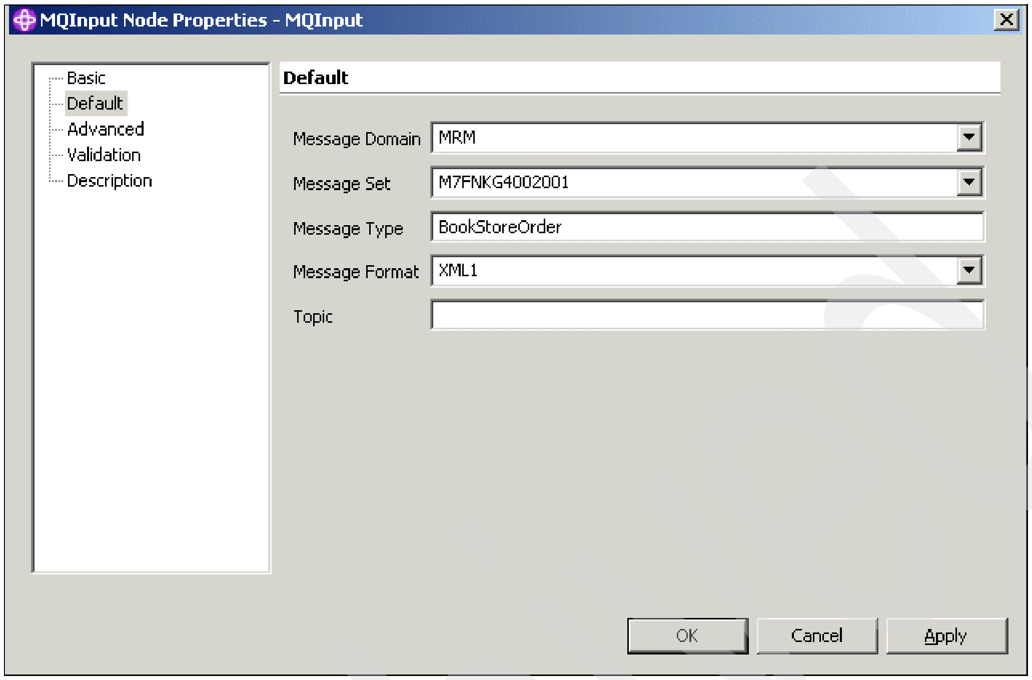


*Figure 4-23 Modified BookstoreFlow message flow*

## Editing the MQInput node properties

To add information about the Bookstore message that is set to the MQInput node:

1. In the Message Flow Editor, right-click the MQInput node, then click **Properties**.
2. On the Default page of the Properties dialog box, edit the properties as shown in Figure 4-24 on page 118:
   a. From the Message Domain list, select **MRM**.
   b. From the Message Set list, select the unique identifier of the message set that you created. (The value shown in the figure is different from the identifier that you are using.)
   c. In the Message Type field, type `BookStoreOrder`.
   d. From the Message Format list, select **XML1**.
3. Click **OK** to close the Properties dialog.
4. Save BookstoreFlow.msgflow.



Default

Message Domain: MRM
Message Set: M7FNKG4002001
Message Type: BookStoreOrder
Message Format: XML1
Topic:

*Figure 4-24 Editing the MQInput node properties to specify message set details*

## Configuring the Mapping node

The Mapping node enables you to drag and drop fields of the input message, BookStoreOrder, onto fields of the output message, ShippingDetails, to map the content of the input message onto the output message.

To configure the Mapping node:

1. In BookstoreFlow.msgflow, right-click the Mapping node, then click **Open Mappings**. A mappings file called BookstoreFlow_Mapping.mfmap is created. BookstoreFlow_Mapping.mfmap automatically opens in the Mapping editor (Figure 4-25 on page 119).

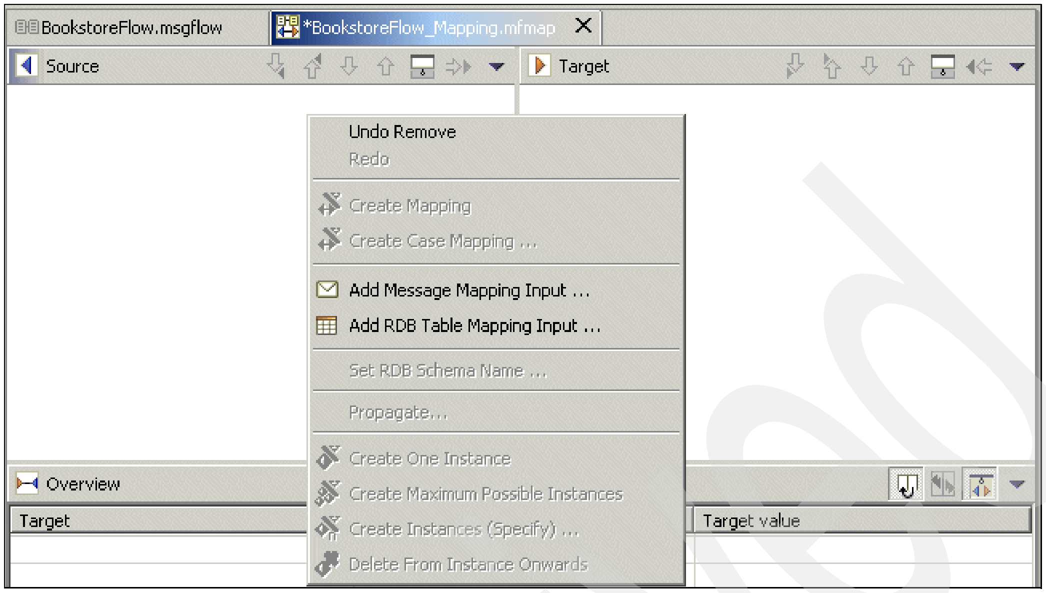
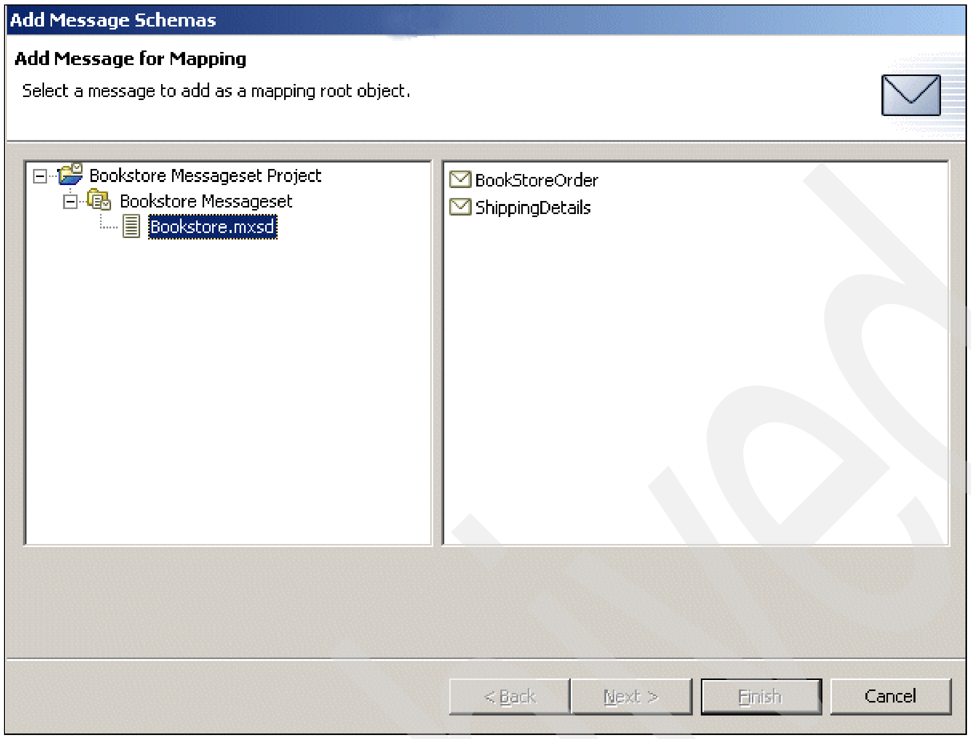

BookstoreFlow.msgflow
*BookstoreFlow_Mapping.mfmap
Source
Target
Undo Remove
Redo
Create Mapping
Create Case Mapping ...
Add Message Mapping Input ...
Add RDB Table Mapping Input ...
Set RDB Schema Name ...
Propagate...
Create One Instance
Create Maximum Possible Instances
Create Instances (Specify) ...
Delete From Instance Onwards
Overview
Target
Target value

*Figure 4-25 Adding the input message definition to BookstoreFlow_Mapping.mfmap*

2. In the Mapping editor, add the message mapping input, which is the message definition of the input message:

   a. In the Mapping editor, right-click the Source pane, then click **Add Message Mapping Input**. This opens the Add Message Schemas wizard.

   b. In the left pane of the Add Message Schemas wizard, expand **Bookstore Messageset Project** → **Bookstore Messageset**, then click **Bookstore.mxsd** to display the two messages, BookStoreOrder and ShippingDetails, in the right pane of the wizard (Figure 4-26 on page 120).


Add Message Schemas

Add Message for Mapping

Select a message to add as a mapping root object.

- Bookstore Messageset Project
  - Bookstore Messageset
    - Bookstore.mxsd

BookStoreOrder
ShippingDetails

*Figure 4-26 Selecting the message definition to use as input to the message mapping*

c. In the right pane of the wizard, click **BookStoreOrder,** then click **Next**.

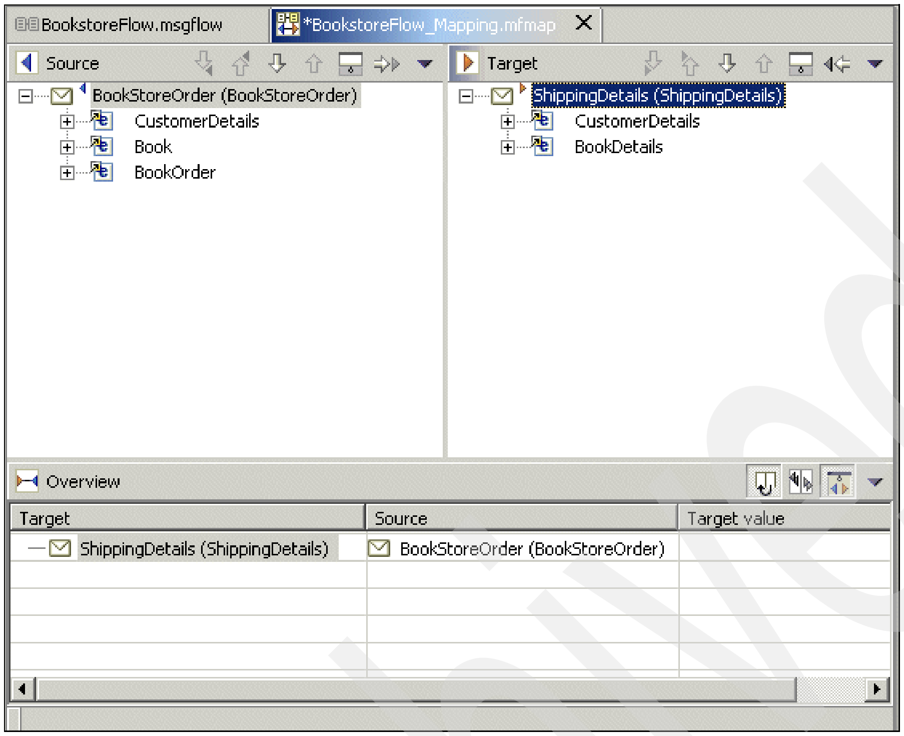
d. Select the **Bookstore Messageset Project**, then click **Finish**. The BookStoreOrder message elements are added to the Source pane of the Mapping editor.

3. In the Mapping editor, add the message mapping output (the message definition of the output message):

   a. In the Mapping editor, right-click the Target pane, then click **Add Message Mapping Output**. The Add Message Schemas wizard opens.

   b. In the left pane of the Add Message Schemas wizard, expand **Bookstore Messageset Project** → **Bookstore Messageset**, then click **Bookstore.mxsd** to display the two messages, BookStoreOrder and ShippingDetails in the right pane of the wizard.

   c. In the right pane of the wizard, click **ShippingDetails** (Figure 4-27 on page 121), then click **Next**.



| Target | Source | Target value |
|---|---|---|
| ShippingDetails (ShippingDetails) | BookStoreOrder (BookStoreOrder) | |

*Figure 4-27 The input and output message definitions in the BookstoreFlow mapping file*

d. Select the **Bookstore Messageset Project**, then click **Finish**. The BookStoreOrder message elements are added to the Target pane of the Mapping editor.

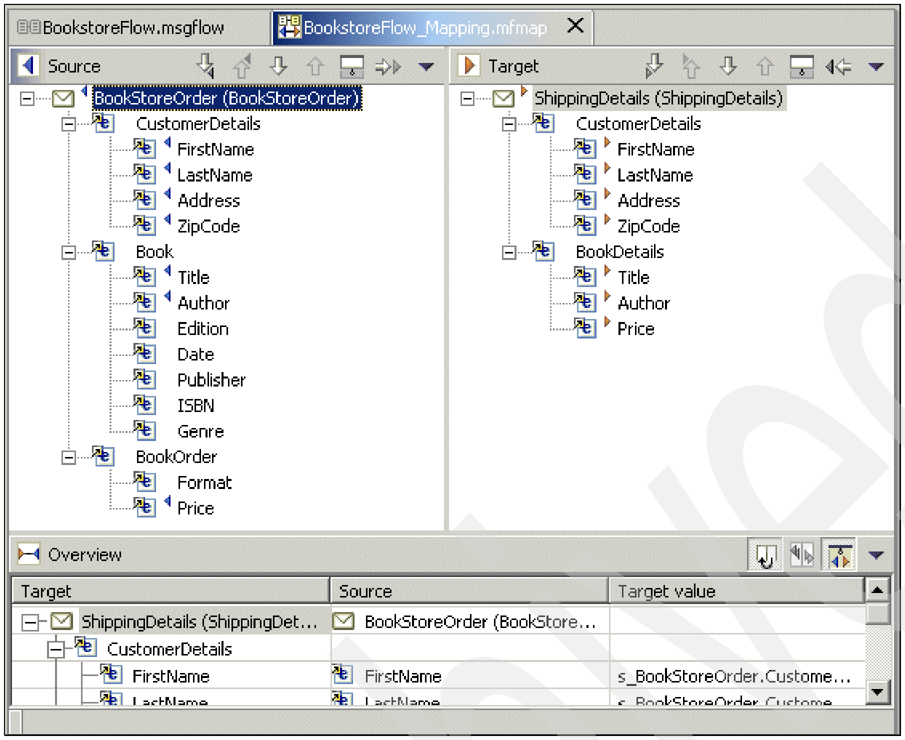
4. Map the fields of the input message onto the output message:

   a. In the Mapping editor, expand all the elements of the messages in the Source and Target panes.

   b. In the Source pane, click and drag the FirstName element to the FirstName element in the Target pane. Small arrowheads appear next to the elements that you have mapped.

   c. Repeat step 4b for each element in the ShippingDetails message. The Overview view, below the Mapping editor, shows a summary of the mappings that you have created (Figure 4-28 on page 122).

5. Save BookstoreFlow_Mapping.mfmap.



| Target | Source | Target value |
|---|---|---|
| ShippingDetails (ShippingDet… | BookStoreOrder (BookStore… | |
| CustomerDetails | | |
| FirstName | FirstName | s_BookStoreOrder.Custome… |
| LastName | LastName | s_BookStoreOrder.Custome |

*Figure 4-28 Mapping the fields of the input message on to the fields of the output message*

You have now configured the BookstoreFlow message flow to use the Bookstore message definition in Bookstore Messageset to parse the input message. The Mapping node in the BookstoreFlow message flow creates an output message to confirm the customer's order by taking the relevant data from the input message to use in the output message.

The final step before you can test the message flow is to create new input and output queues for the BookstoreFlow message flow.

## Creating new input and output queues for the message flow

When you deployed the original version of the BookstoreFlow message flow, you configured the RAD server to deploy only a message flow. Therefore, when you deploy the new version of the BookstoreFlow message flow, you must create a new server configuration to include the Bookstore Messageset as well as the BookstoreFlow message flow. However, this means that both versions of the BookstoreFlow message flow will coexist in the broker, each in a different execution group. (Each new server configuration creates a new execution group.)

Both versions of the message flow will try to take the input message from the IN queue at the same time.

To prevent conflict between the two versions of the BookstoreFlow message flow, create two new queues for the new version of the BookstoreFlow message flow and configure the MQInput node and the MQOutput node to interact with the new queues:

1. Create two new WebSphere MQ local queues called `MRM_IN` and `MRM_OUT`.
2. Edit the properties of the MQInput node so that it gets messages from the queue called MRM_IN.
3. Edit the properties of the MQOutput node so that it puts messages onto the queue called MRM_OUT.

The new version of the BookstoreFlow message flow is now configured.

## 4.3.3 Testing the modified Bookstore message flow application

To test the Bookstore message flow application with the message set:

1. Deploy Bookstore Messageflow Project and Bookstore Messageset Project:
   a. In the Resource Navigator view, right-click **Bookstore Messageflow Project**, then click **Run On Server**.
   b. In the Server Selection wizard, click **Create a new server**.
   c. Work through the Create a New Server wizard:
      i. Call the new execution group `Bookstore2`.
      ii. On the final page of the wizard, when you select **Bookstore Messageflow Project**, the wizard automatically selects Bookstore Messageset Project.

   The BookstoreFlow message flow and Bookstore Messageset are both deployed to an execution group called Bookstore2 in the broker.
2. Put a test message through the BookstoreFlow message flow:
   a. In the Resource Navigator view, double-click **XML_test_msg.enqueue**, which you created earlier in this chapter, to open it in the Enqueue editor.
   b. In the Enqueue editor, change the queue name to `MRM_IN`, then click **Write To Queue** to put the message on the MRM_IN queue.
   c. Open the Dequeue editor.
   d. In the Dequeue editor, change the queue name to `MRM_OUT`, then click **Read From Queue**. The Dequeue editor displays the output message that the BookstoreFlow message flow generated from the input message.

In this chapter, you have created a simple message flow, BookstoreFlow, which has three nodes: MQInput, Compute, and MQOutput. You tested the BookstoreFlow message flow using a self-defining XML message. You then created a message set, Bookstore Messageset, which externally defined the structure of the XML message in a message set definition called Bookstore. You adapted the BookstoreFlow message flow to use the Bookstore Messageset to parse the input message and to generate the output message, and tested the application.

In terms of the original scenario, at the online bookstore, a message is generated to confirm the shipping details of the customer's order.

The final section of this chapter looks briefly at how you can extend the processing capability of a message flow to perform complex manipulations, transformations, and routing of messages.

## 4.4 Extending the processing of message flows

The original version of the BookstoreFlow message flow performs some very basic processing using just three nodes:

- The *MQInput node* gets an input message from the IN queue.
- The *Compute node* builds an output message that contains the same content as the input message.
- The *MQOutput node* puts the output message on the OUT queue.

The modified version of the BookstoreFlow message flow performs slightly more complex processing using a Mapping node instead of the Compute node.

As you can see in the Node palette in the Message Flow Editor, there is a large selection of other built-in nodes available to use in message flows. This section describes which nodes to use to perform certain processing functions.

Note that only the input and output nodes are available in WebSphere Business Integration Event Broker.

For examples of how to use some of the built-in nodes, see the Samples Gallery in the WebSphere Business Integration Message Brokers Help system. Each of the following sections suggests relevant samples to look at in the Samples Gallery.

## 4.4.1 Providing input to a message flow

So that business applications can pass messages to the message flow, the message flow must have at least one input node. The BookstoreFlow message flow contains an MQInput node because the message flow gets the messages from a WebSphere MQ queue.

The type of input node that is used in a message flow depends on the type of application that sent the message to the broker:

- If the message flow gets the messages from the business application by means of a WebSphere MQ queue, use an *MQInput node*.
- If the messages are sent to broker by an MQe application, use an *MQeInput node*.
- If the messages are sent by a telemetry device, use a *SCADAInput node*.
- If the messages are sent by a Web services client, use an *HTTPInput node*.
- If the messages are sent by a JMS or multicast application, use a *Real-timeInput node* or a *Real-timeOptimizedFlow node*.
- If the message flow is to be reused as part of one or more other message flows, use an *Input node*.

Figure 4-29 shows all of the input nodes that are available in WebSphere Business Integration Message Broker Version 5.0. The node called Input node is slightly different from the others because it is used exclusively in subflows. Subflows are message flows that are embedded in other message flows. The main benefit of subflows is that you can maintain them independently of the main message flow. Also, by modularizing part of a message flow as a subflow, you can easily reuse that part of the message flow logic in another message flow application.

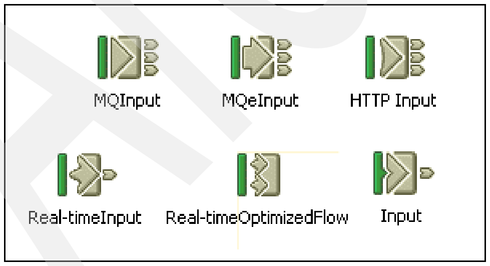


*Figure 4-29 The built-in input nodes*

Most of the samples in the Samples Gallery demonstrate the MQInput node. The XML_BuildReplyMessage subflow in the Airline Reservations sample and the subflow in the Error Handler sample demonstrate the Input node. The Scribble sample demonstrates the Real-timeInput node.

## 4.4.2 Getting output from a message flow

So that business applications can receive messages from the message flow, the message flow must have at least one output node. The BookstoreFlow message flow contains an MQOutput node because the message flow puts the messages on a WebSphere MQ queue.

The output node that is used in a message flow depends on the type of application that receives the message from the broker:

- If the message flow puts the message to the application by means of a WebSphere MQ queue, use an *MQOutput node*.
- If you want the message flow to put different messages on different WebSphere MQ queues, depending on the queue name in the Reply-to queue field of the message header WebSphere MQ, use an *MQReply node*. Using the MQReply node means that the name of the output queue is not hardcoded in the message flow.
- If the messages are received by a telemetry device, use a *SCADAOutput node*.
- If the messages are received by a Web services client, use an *HTTPReply node*.
- If the message flow interacts with a Web services client, use an *HTTPRequest node*.
- If the messages are received by a JMS or multicast application, use a *Real-timeOptimizedFlow node*. This node performs as both an input and an output node.
- If the message flow is to be reused as part of one or more other message flows, use an *Output node*.
- If you want to distribute the messages, using Publish/Subscribe, to many unknown applications that have subscribed to receive the data, use a *Publication node*.

All of the output nodes that are available in WebSphere Business Integration Message Broker Version 5.0 are shown in Figure 4-30 on page 127. The node called Output node, like the Input node, is slightly different from the others because it is used exclusively in subflows. The Publication node is also different because the destination of the messages depends entirely on whether the receiving applications have subscribed to the information that the messages contain. For more information about Publish/Subscribe, see the WebSphere Business Integration Message Broker help system.

You can also define your own output node if none of the built-in output nodes is suitable for the protocol or transport type that your business application uses.

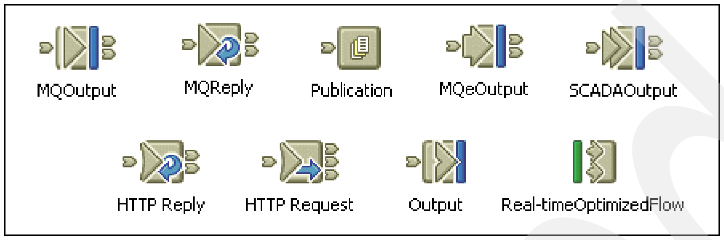


*Figure 4-30 The built-in output nodes*

Most of the samples in the Samples Gallery demonstrate the MQOutput node. The XML_BuildReplyMessage subflow in the Airline Reservations sample and the subflow in the Error Handler sample demonstrate the Output node. The XML_FlightQueryReply message flow in the Airline Reservations sample demonstrates the MQReply node. The SurfWatch sample (one of the Getting Started samples), the Soccer Results sample, and the Scribble sample all demonstrate the Publication node.

### 4.4.3 Manipulating and transforming the content of messages

If your business is large, it is likely that while the business has developed, many business applications have been developed. If the applications have been developed to run on different platforms and they store data in different formats, it is difficult to send data between them. To help solve this problem, you can develop message flows that manipulate, or even change the format of, the messages that the broker receives from one application before it passes them to another application.

- To manipulate the content of a message, for example, by modifying the fields that are included in the output message, or by retrieving information from a database and including it in the output message, use a *Compute node*. You must program the node using ESQL. The Compute node does not require that you have externally defined the messages in a message set.
- To create a new message from all or some of the data in the input message by mapping the input fields onto the output fields, use the *Mapping node* (for example, if one application requires that the last name is before the first name but the other application requires that the first name is before the last name). If you have externally defined the message structure in a message set, you can easily produce simple mappings using the mappings editor.

- To create a partial copy of the input message by extracting specific fields from the input message, use the *Extract node*. As with the Mappings node, you must have defined the message structure externally in a message set.
- To interact with a database in any way, such as inserting data into a database that another message flow can then extract and use, use a *Database node*. You can use the Database node whether or not you have predefined the message in an external message set. You must program the manipulations using ESQL.
- To interact more easily in more specific ways with a database, use the following nodes, which can manipulate only messages that have been predefined in an external message set:
  - The *DataDelete node* deletes one or more rows from a database table.
  - The *DataInsert node* inserts one or more rows of data into a database table.
  - The *DateUpdate node* modifies the data in one or more rows of a database table.
- To store all or part of messages in a database to keep a record of the messages that the broker receives, use the *Warehouse node*. You can use the Warehouse node only if the message has been predefined in an external message set.

You can also define your own nodes to manipulate the messages that your business applications use if none of the built-in nodes is suitable.

Figure 4-31 shows all of these nodes.

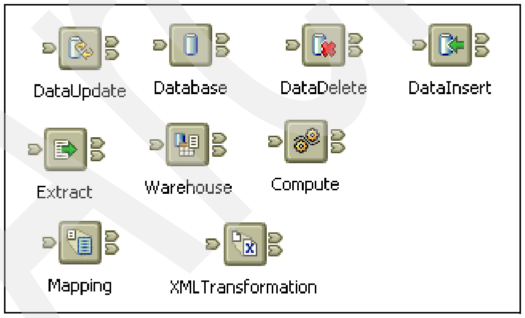


*Figure 4-31 The built-in message manipulation and transformation nodes*

All of the main message flows in the Airline Reservations sample demonstrate the Compute node. The XML_Reservation message flow in the Airline Reservations sample and the main message flow in the Error Handler sample demonstrate the Database node.

## 4.4.4 Dynamically routing messages

Your business might require that financial transactions of more than a specified amount must be recorded separately for audit reasons, or you might want different messages to be routed to different applications depending on their content. If so, you can program message flows to dynamically route the messages to different branches of the message flow according to whether the message content matches defined rules.

- To verify that the structure of a message is correct for subsequent processing, use the *Check node*.
- To perform a Boolean filter on a message so that if it matches the rule the message takes one route through the message flow, and if it does not match the rule the message takes another route through the message flow, use the *Filter node*.
- To force messages to be processed in one way and then processed in a second way, use the *FlowOrder node*.
- To define a number of alternative destinations to which messages can be sent, depending on the message header, use a *Label node* for each destination, and a *RouteToLabel node* to determine which Label node a message must be sent to.
- To change the message properties before the message is parsed again by nodes downstream in the message flow, use the *ResetContentDescriptor node*.

Figure 4-32 shows all of the nodes that can be used for dynamically routing messages.

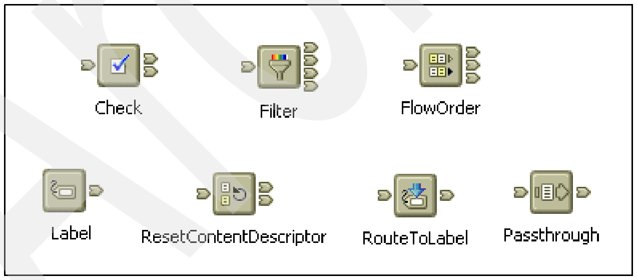


*Figure 4-32 The built-in dynamic routing nodes*

The XML_BuildReplyMessage subflow in the Airline Reservations sample, the main message flow and the subflow in the Error Handler sample demonstrate the Filter node. The XML_PassengerQuery message flow in the Airline Reservations sample demonstrates the RouteToLabel and Label nodes.

## 4.4.5 Collating multiple responses to a request

If you need the message flow to look up several pieces of information but the information is confidential, or if the information is held in several different places, you can program the message flow to request each piece of information separately and prepare a response only if all of the information is successfully acquired. To do this, use the following nodes together in a message flow:

- The *AggregateControl node* issues multiple requests, one request for each piece of information required.
- *AggregateRequest nodes* record each request.
- The *AggregateReply node* collates all of the responses and, if it receives all of the responses within a certain time, the node issues a response to the original request.

Figure 4-33 shows the three nodes that are used for collating multiple responses to a request.

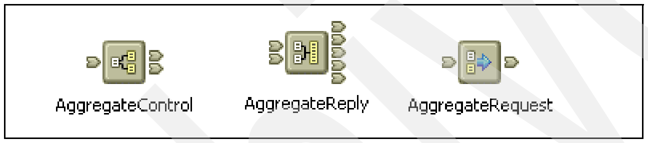


*Figure 4-33 The built-in aggregation nodes*

The XML_FlightQuery message flows in the Airline Reservations sample demonstrate the AggregateControl node, the AggregateReply node, and the AggregateRequest node.

## 4.4.6 Error handling

So that you can diagnose and fix processing errors in message flows, use the following nodes:

- To generate trace records of the processing at a specific point in the message flow, use a *Trace node*.
- To control the way that a message flow deals with messages that it cannot process (for example, to remove the message from message flow to prevent it blocking other messages), use the *TryCatch node*.
- To force the message flow to throw an exception if an error occurs during processing, use a *Throw node*.

Figure 4-34 on page 131 shows all of the nodes that can be used for error handling and reporting.

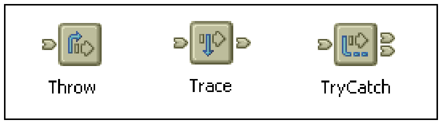


*Figure 4-34 The built-in error handling nodes*

The XML_BuildReplyMessage subflow in the Airline Reservations sample and the subflow in the Error Handler sample demonstrate the Throw node. The subflow in the Error Handler sample also demonstrates the TryCatch node. The XML_Reservation message flow and the XML_CancelReservation message flow in the Airline Reservations sample demonstrate the Trace node.

## 4.4.7 Getting more information about the built-in nodes

For more detail about how and when to use each of the nodes, see the WebSphere Business Integration Message Broker and WebSphere Business Integration Event Broker product documentation. To try out the different nodes to see which nodes are most useful to you, explore the samples in the WebSphere Business Integration Message Brokers Samples Gallery.

5

# Administration

This chapter provides an overview of the administration of the runtime environment for WebSphere Business Integration Message Brokers using the Message Brokers Toolkit and the command line interface. The following topics are discussed in this chapter:

- Creating a broker domain
- Deploying message flow applications
- Publish/subscribe

## 5.1 Creating a broker domain

Once broker applications in the form of message flows and message sets have been developed, then a broker domain has to be created in order to test and utilize these flows and any associated message sets. A broker domain represents the Message Broker's runtime environment. A broker domain contains a single Configuration Manager with associated brokers. It may also contain a User Name Server if authentication is used for publish/subscribe.

Information about the broker domain is stored in the Configuration Manager's database, which is known as the configuration repository, but a reference to the broker domain is made in the Message Brokers Toolkit where the domain is administered.

The first stage of creating a broker domain is to create the required components on the command line. The most basic broker domain configuration is a Configuration Manager and a single broker. This is the configuration that the "default configuration wizard" also known as the "Getting Started Wizard" creates, as mentioned in Chapter 3, "Installation" on page 23, Section3.4, "Installing product fix packs" on page 52.

When the default configuration wizard is used to create an initial configuration, then the end result is a simple broker domain with a Configuration Manager and a broker (by default WBRK_BROKER). The wizard also creates the resources required by these two components, that need to be created manually if the wizard is not used.

If the default configuration wizard has been used to create the initial broker domain configuration then the existing broker domain connection can be removed by deleting the LocalProject directory from the Message Brokers Toolkit. This only deletes the reference to the broker domain in the toolkit, and not delete any of the data from the configuration repository. Once this has been removed, then the connection to the broker domain can be re-created, instructions in section 5.1.2 below.

### 5.1.1 Manually creating a basic Message Broker configuration

This section gives instructions for creating the components and resources for a basic broker domain using manual methods.

#### WebSphere MQ resources

For the most basic broker domain configuration, the Configuration Manager and a single broker can share the same queue manager. This makes the setting up of the WebSphere MQ resources much simpler, as there is no requirement for the setting up of channels between multiple queue managers. It is also the case that the User Name Server, if required, could use this same queue manager. It is recommended, however, in a production environment that each component has its own queue manager defined for performance reasons.

A WebSphere MQ queue manager can be created on the command line or by using the WebSphere MQ Services and WebSphere MQ Explorer programs. WebSphere MQ Services is used in this example, as this program is also used to create a listener.

Note that the commands that are used to create a Configuration Manager, a broker, or a User Name Server creates a queue manager if one does not already exist, but a listener is not created automatically.

Create a queue manager using WebSphere MQ Services:

1. Select **Start** → **Program Files** → **IBM WebSphere MQ** → **WebSphere MQ Services.**
2. Right-click **WebSphere MQ Services (local)**.
3. Select **New** → **Queue Manager** from the context menu.
4. In the Create Queue Manager wizard, enter `WBRK_QM` for the queue manager name.
5. Leave the other settings as default and click **Next** until step 4 of the wizard.

The last step of the wizard enables the creation of a listener service for TCP/IP communications. A port number for the listener must be specified. This port number is used for communication between the Message Brokers Toolkit and the Configuration Manager.

The default value for the port number on WebSphere MQ is 1414, and this is the value that is used if the default configuration wizard in the toolkit is used to create the broker domain. If a default queue manager on WebSphere MQ has been configured using the First Steps tool, for example, then a listener may be present already on port 1414, so a different port number must be configured for this queue manager. The exact range of the port numbers that are available to the user is dependent on the hardware and software environment of the system, and different port numbers may be in use. However, a variety of applications and commands can be used for viewing port usage on a machine, such as the **`netstat`** command. Use the command **`netstat -a`** to display a list of active TCP connections and ports on a Windows machine. Choose an alternative port to those that are currently in use.

6. Enter a value for the Port number.
7. Click **Finish**.

## Database resources

The Configuration Manager and any brokers store configuration data in a database. In the simplest broker domain configuration, DB2 is used for both the Configuration Manager database and the broker database. A single database could be used for both the configuration respository and the broker tables, but in this example a separate database is used for each.

As with the WebSphere MQ resources, the databases can be created on the command line; however, in our example, the DB2 Control Center is used to create the databases.

1. Create the database for the Configuration Manager:
   a. Open the DB2 Control Center. The exact location of this program depends on the DB2 version and the installation, but you might try: **Start** → **Program Files** → **IBM DB2** → **Control Center**.
   b. Navigate to the Databases folder in the DB2 Control Center.
   c. Right-click **Databases**, and select **Create** → **Database Using Wizard** from the context menu.
   d. The first database to be created is the Configuration Manager database, so enter the name `WBRKCMDB` in the Database name field.
   e. Leave the default settings in the other fields and click **Finish**.

   This initiates the creation of a new database called WBRKCMDB, which will be used later in the creation of the Configuration Manager.
2. Create a broker database:
   a. Repeat the process in step 1 to create a second database called `WBRKBKDB`, which will be used for the creation of a broker.
   b. After this database has been created, close the DB2 Control Center.

The final step in creating the database resources is to create an Open Database Connectivity (ODBC) connection to the broker database. ODBC connections enable Message Broker to connect and exchange data with the DB2 database that has just been created. An ODBC connection does not have to be set up for the Configuration Manager, as this uses Java Database Connectivity (JDBC) to connect to its database. ODBC connections are set up in the ODBC Data Source Administrator in Windows.

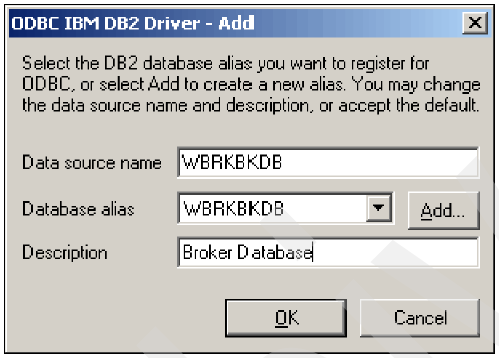
3. Create an ODBC connection for the broker database:
   a. Open **Control Panel** → **Administrative Tools** → **Data Sources (ODBC)**. (The location and name of the ODBC Administrator may vary between systems.)

b. In the ODBC Data Source Administrator, select the **System DSN** tab, and click **Add** to add the `WBRKBKDB` database to the list of databases on the system.

c. Select **IBM DB2 ODBC DRIVER** from the list of driver options that are displayed in the Create New Data Source dialog.

d. Click **Finish**.

e. From the pull-down list, select the **WBRKBKDB** database in the Database alias field (Figure 5-1).



Select the DB2 database alias you want to register for ODBC, or select Add to create a new alias. You may change the data source name and description, or accept the default.

Data source name: WBRKBKDB
Database alias: WBRKBKDB
Description: Broker Database

*Figure 5-1 ODBC Data Source Administrator: Add*

f. Enter the same name in the Data source name field. A description can be added to indicate the purpose of the database, but this is optional.

g. Click **OK** to complete adding an ODBC reference for this database.

h. Close the ODBC Data Source Administrator.

## Creating a Configuration Manager

After the database and WebSphere MQ resources have been created, the WebSphere Business Integration Message Brokers components can be created. There is no specific order in which the components have to be created, although there is more significance with order in the starting and deleting of components. The creation of the components in this section must be carried out on a command line, as no facility is provided to create the components with the Message Brokers Toolkit.

Create a Configuration Manager using the command line:

1. Select **Start** → **Run**.
2. Type `cmd` in the Open field.

3. Use the following command syntax to create a Configuration Manager on the command line:

   ```
   mqsicreateconfigmgr -i userid -a password -q WBRK_QM -n WBKRCMDB
   ```

   The necessary parameters are:

   - `-i` is a user ID belonging to the mqbrks group.
   - `-a` is the user ID’s matching password.

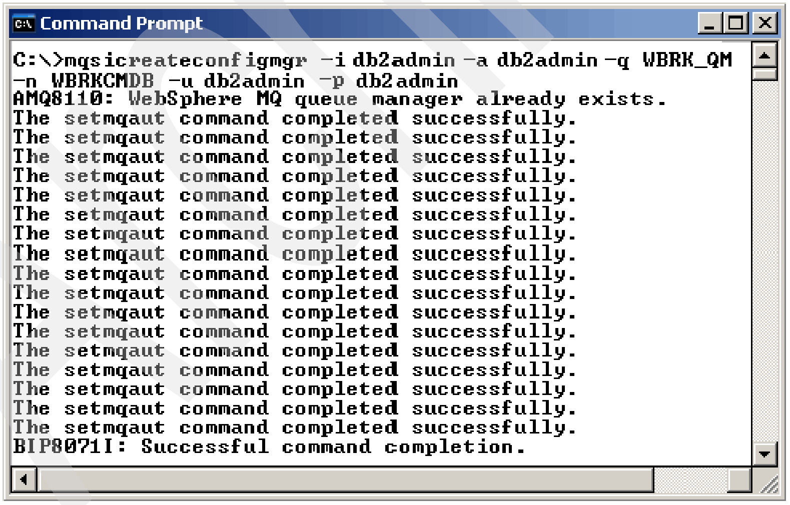
   If a different user ID is required for access to the database, then the following syntax and parameters should be used:

   ```
   mqsicreateconfigmgr -i userid -a password -q WBRK_QM -n WBKRCMDB -u
   dbuserid -p dbpassword
   ```

   The additional parameters are:

   - -u is the database user ID
   - -p is the database password

   The `-q` parameter in each of these examples indicates which WebSphere MQ queue manager will be used for the Configuration Manager communication, and the `-n` is the name of the database in which the Configuration Manager stores topology data. If the queue manager that is specified here on the command line does not exist, then the **`mqsicreateconfigmgr`** command can be used to create it. The results of successfully creating a Configuration Manager can be seen in Figure 5-2.



```
C:\>mqsicreateconfigmgr -i db2admin -a db2admin -q WBRK_QM
-n WBRKCMDB -u db2admin -p db2admin
AMQ8110: WebSphere MQ queue manager already exists.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
BIP8071I: Successful command completion.
```

*Figure 5-2 Successful creation of a Configuration Manager*

## Creating a broker

Creating a broker on the command line is similar to creating a Configuration Manager and requires similar parameters.

Create a broker using the command line:

1. Use the following syntax in a command window to create a broker called `WBRK_BROKER`:

   ```
   mqsicreatebroker WBRK_BROKER -i userid -a password -q WBRK_QM -n
   WBRKBKDB
   ```

   - `-i` is a user ID belonging to the mqbrks group.
   - `-a` is the matching password.

   Other parameters are available, but are not required for a simple configuration.

   If a different user ID is required for access to the database, then the following syntax and parameters should be used:

   ```
   mqsicreatebroker WBRK_BROKER -i userid -a password -q WBRK_QM -n
   WBKRBKDB -u dbuserid -p dbpassword
   ```

   - `-u` is the database user ID.
   - `-p` is the database password.

The broker name is the parameter after the **`mqsicreatebroker`** command. (In this example, `WBRK_BROKER` is used.) The `-q` parameter indicates which WebSphere MQ queue manager the broker will use. For a basic configuration, this should be the same queue manager as specified for the Configuration Manager, but if a different queue manager name was used, then a new queue manager would be created by the **`mqsicreatebroker`** command if required. The `-n` parameter is the name of the database that the broker will use to store configuration data.

After the command has completed, a BIP8071 success message should be displayed.

## Starting the components

The final task in the creation of the basic configuration is to start the Configuration Manager and the broker.

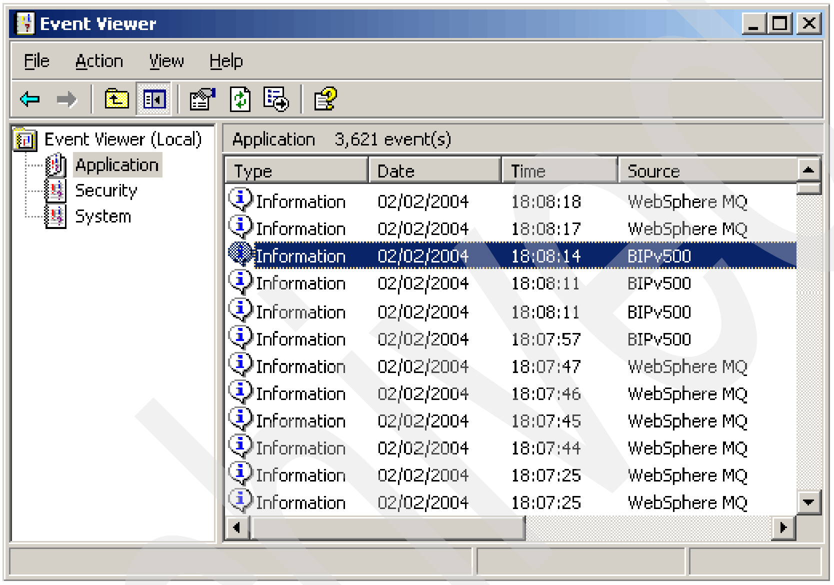
1. Start the Configuration Manager by typing this command on the command line:

   ```
   mqsistart configmgr
   ```

   This success message should be displayed after the command has been run:

   ```
   BIP8096I: Successful command initiation, check the system log to ensure
   that the component started without problem and that it continues to run
   without problem.
   ```

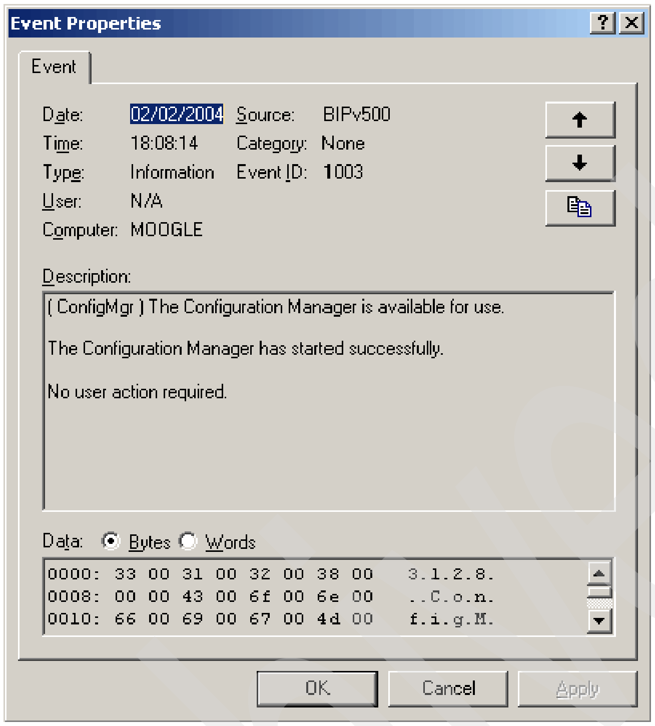
2. Verify that the Configuration Manager has started successfully and become available for use by checking for BIPv500 messages in the Windows Event Viewer. These messages are generated by WebSphere Business Integration Message Brokers runtime. Starting the Windows Event Viewer (Figure 5-3) is dependent on the version of Windows that is being used, but typically it can be launched from the Administrative Tools section of the Control Panel, or in the Computer Management tool found in the Control Panel.



| Type | Date | Time | Source |
|---|---|---|---|
| Information | 02/02/2004 | 18:08:18 | WebSphere MQ |
| Information | 02/02/2004 | 18:08:17 | WebSphere MQ |
| Information | 02/02/2004 | 18:08:14 | BIPv500 |
| Information | 02/02/2004 | 18:08:11 | BIPv500 |
| Information | 02/02/2004 | 18:08:11 | BIPv500 |
| Information | 02/02/2004 | 18:07:57 | BIPv500 |
| Information | 02/02/2004 | 18:07:47 | WebSphere MQ |
| Information | 02/02/2004 | 18:07:46 | WebSphere MQ |
| Information | 02/02/2004 | 18:07:45 | WebSphere MQ |
| Information | 02/02/2004 | 18:07:44 | WebSphere MQ |
| Information | 02/02/2004 | 18:07:25 | WebSphere MQ |
| Information | 02/02/2004 | 18:07:25 | WebSphere MQ |

*Figure 5-3 Windows Event Viewer*

The contents of a message can be viewed by double-clicking the message name. The specific message that indicates that the Configuration Manager is started and available for use is a message with an Event ID of `1003`, as shown in Figure 5-4 on page 141. For more about error messages in the Windows Event Viewer, refer to 6.1.2, “Windows Event Viewer” on page 171.



Date: 02/02/2004 Source: BIPv500
Time: 18:08:14 Category: None
Type: Information Event ID: 1003
User: N/A
Computer: MOOGLE

Description:
( ConfigMgr ) The Configuration Manager is available for use.

The Configuration Manager has started successfully.

No user action required.

Data: [✓ Bytes] Words

```
0000: 33 00 31 00 32 00 38 00   3.1.2.8.
0008: 00 00 43 00 6f 00 6e 00   ..C.o.n.
0010: 66 00 69 00 67 00 4d 00   f.i.g.M.
```

*Figure 5-4 Application log message: Configuration Manager is available for use*

If this message is not present, the Configuration Manager is not available for use, even if it may have started correctly. The Event Viewer may display error messages that give diagnostic information to help track down the cause of the problem. For more information about diagnosing problems, see 6.1.2, “Windows Event Viewer” on page 171.

3. The next step is to start the broker by typing this command on the command line:

```
mqsistart WBRK_BROKER
```

This assumes that the broker name is `WBRK_BROKER`. If a different broker name has been used, use it instead of `WBRK_BROKER`.

This success message should be displayed after the command has been run:

```
BIP8096I: Successful command initiation, check the system log to ensure
that the component started without problem and that it continues to run
without problem.
```

If a value has not been set for the number of license units using the **mqsisetcapacity** command after installation, then the following warning message will be displayed before the success message:

```
BIP8810W: Insufficient license units.
The purchased processor allowance (0) is less than the number of
processors (1) in this machine. Ensure sufficient license units have
been purchased and use the WebSphere Business Integration Message
Brokers mqsisetcapacity command to set the purchased processor allowance
for this installation. For more information refer to the online help.
```

4. Use the following instructions to set the number of license units for the machine if that error is displayed in this format, replacing *n* with the number of processors on the machine:

   ```
   mqsisetcapacity -c n
   ```

   Sufficient licenses must have been purchased to cover the number of processors on each machine. The following example is for a machine with 2 processors:

   ```
   mqsisetcapacity -c 2
   ```

5. Check the Windows Event Viewer for a BIPv500 message with an EventID of `2001` to confirm that the broker started successfully.

If for any reason the Configuration Manager or broker failed to be created or start successfully, refer to 6.1.2, "Windows Event Viewer" on page 171 for information about how to perform problem determination.

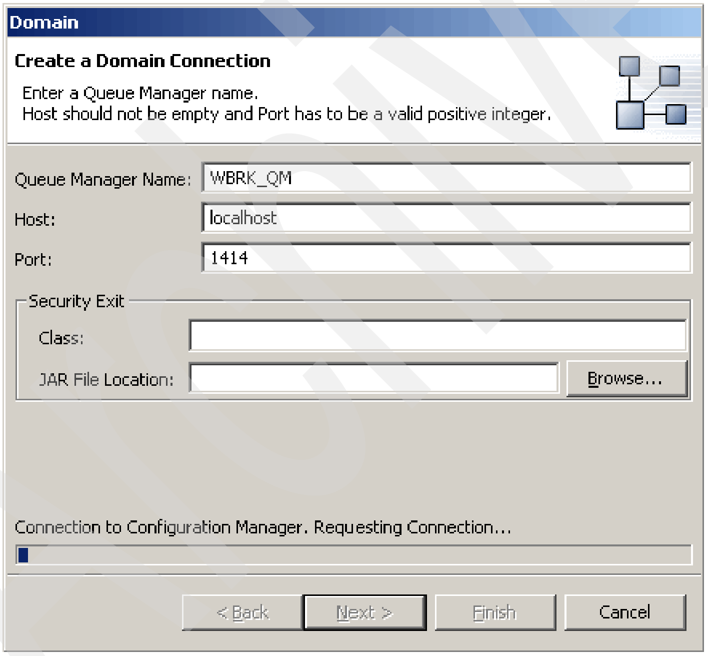
## 5.1.2 Creating a domain connection

This section assumes that the basic components of a broker domain, a Configuration Manager and broker, have been created and started on the system. Refer to Chapter 3, "Installation" on page 23 and 5.1, "Creating a broker domain" on page 134 for instructions to create and start these components for a basic configuration. If the default configuration wizard has been used, then a domain connection will have been created, but this can be removed in order to create a domain connection manually if desired by deleting the LocalProject directory from the Message Brokers Toolkit. This only deletes the reference to the broker domain in the toolkit, and does not delete any of the data from the configuration repository.

The domain connection is the reference to a broker domain in the Message Brokers Toolkit. Multiple broker domain connections can be defined in the toolkit to both local and remote Configuration Managers. To use the default configuration components to create a broker domain connection on the local machine:

1. Create a domain connection:
   a. Launch the Message Brokers Toolkit.
   b. Change to the Broker Administration Perspective by selecting **Window** → **Open Perspective** → **Broker Administration**.
   c. From the **File** menu, select **New** → **Domain**.
   d. Enter the Queue Manager name (for example, `WBRK_QM`) in the Domain Connection dialog box.
   e. Click **Next**.

   The Message Brokers Toolkit attempts to communicate with the WebSphere MQ queue manager at this point, and a status bar is displayed in the dialog shown in Figure 5-5.



Enter a Queue Manager name.
Host should not be empty and Port has to be a valid positive integer.

Queue Manager Name: WBRK_QM
Host: localhost
Port: 1414

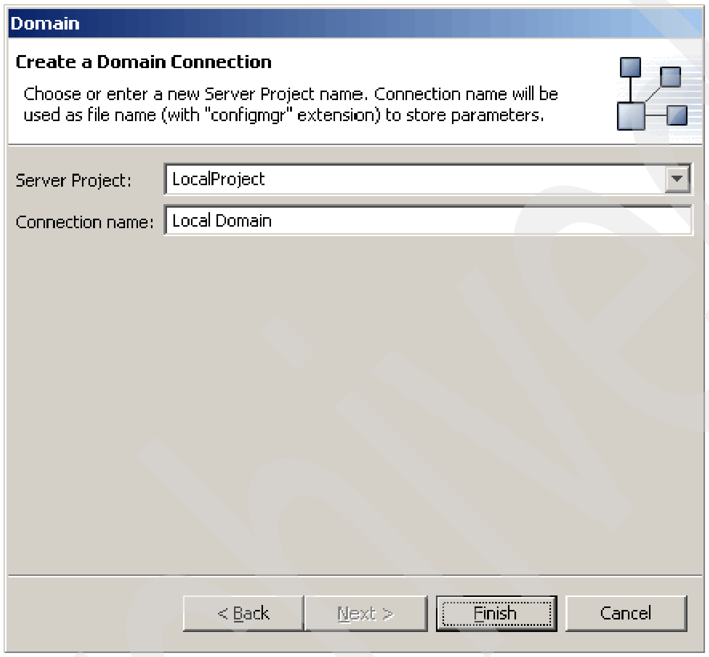
Connection to Configuration Manager. Requesting Connection...

*Figure 5-5 Create a Domain Connection wizard: Queue Manager Configuration*

When a connection has been established, the second page of the wizard is displayed (Figure 5-6). We use the default values from the configuration wizard to complete the dialog:

a. Enter `LocalProject` for the Server Project name.

b. Enter `LocalDomain` for the Connection name.



Domain

**Create a Domain Connection**

Choose or enter a new Server Project name. Connection name will be used as file name (with "configmgr" extension) to store parameters.

Server Project: LocalProject

Connection name: Local Domain

*Figure 5-6 Create a Domain Connection wizard: Select Server Project*

A *project* is an Eclipse term for the logical and physical storage of data resources within the Eclipse Workbench that map to a directory on the file system. There are different types of projects, and only certain resources can be contained in each project type. Resources that are created for Broker Administration must be stored in a Server Project. These files include the connection file, which is created to store the data for the Domain Connection. Other types of projects are introduced in 4.2, "Developing a simple message flow" on page 85.

The Connection name is used to name the file that stores the Configuration Manager connection data for the Domain Connection. Multiple Domain Connections can be created in the Message Brokers Toolkit, and the same Server Project can be used to store them, but the Connection names must differ.

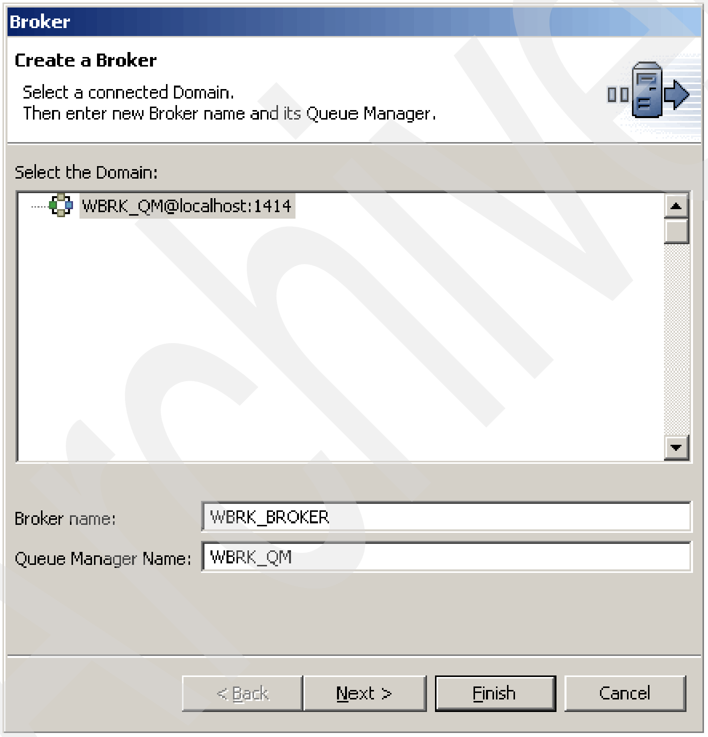
2. To complete the creation of a domain connection, click **Finish**, then click **Yes** to this message:

```
Do you want to create a new Server Project with the name LocalProject?
```

   In the Domains editor at the bottom-left of the screen, the domain connection is now displayed. With the default configuration settings, the connection is displayed as `WBRK_QM@localhost:1414`.

## 5.1.3 Adding a broker to the domain

The next step is to add a broker to the domain using the Broker wizard shown in Figure 5-7. This wizard does not create a broker component; it creates a reference in the domain to a broker that already exists in the system.


Broker

Create a Broker

Select a connected Domain.
Then enter new Broker name and its Queue Manager.

Select the Domain:

WBRK_QM@localhost:1414

Broker name: WBRK_BROKER

Queue Manager Name: WBRK_QM

*Figure 5-7 Create a broker wizard*

To add a broker to the domain, follow these instructions in the Broker Domain Perspective (we use the default broker name and queue manager name):

1. Add a broker to the domain:
   a. Open the Broker Domain Perspective.

b. Select **New** → **Broker** from the **File** menu.

c. Select the Domain to which the broker is to be added. (There will be only one if a simple configuration has been created.)

d. Enter `WBRK_BROKER` for the Broker name.

e. Enter `WBRK_QM` for the queue manager name.

f. Click **Finish**.

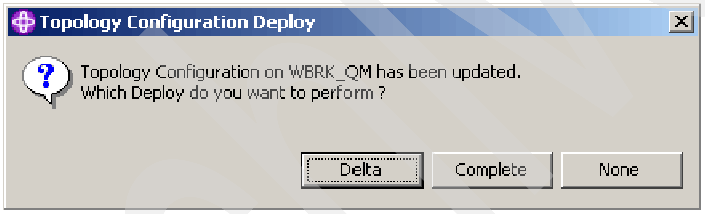
2. After clicking Finish, a Topology Configuration Deploy dialog is displayed for the user to choose the type of deploy to be performed. Figure 5-8 shows the three options: Delta, Complete and None. The topology is the configuration of brokers in the domain:

   – *Delta* deploys only changes to the topology

   – *Complete* deploys the entire configuration

   – *None* results in no deploy being made at that time, so no changes occur in the runtime, although the changes are stored in the toolkit.

   Select **Delta** in the Topology Configuration Deploy dialog box.


Topology Configuration Deploy

Topology Configuration on WBRK_QM has been updated.
Which Deploy do you want to perform ?

*Figure 5-8 Topology Configuration Deploy Selection Dialog Box*

3. A success message pops up when the deploy request is communicated successfully to the Configuration Manager. Click **Details** for more information about the message.

4. To confirm that adding the broker to the domain and the associated topology deploy were successful, check for messages in the Event Log Editor in the Message Brokers Toolkit. This can be accessed in the Domains view of the Broker Administration perspective by double-clicking **Event Log**, as shown in Figure 5-9 on page 147.

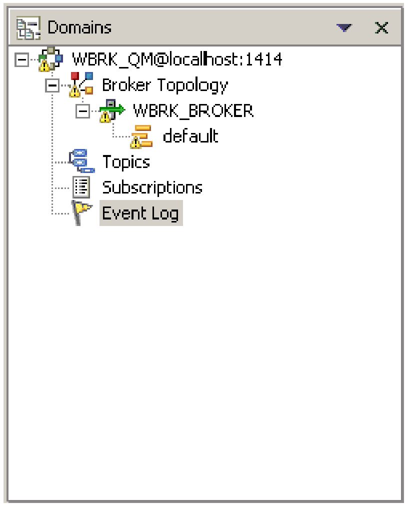


*Figure 5-9 Domains View in the Broker Administration Perspective*

Success messages to look for from the deploy are `BIP2046` and `BIP4045`. The broker is also displayed in the Domains view, together with the default execution group that is created with the broker. In the Alerts view, the warning message `Execution Group is not running` is displayed. This is a normal message to see before any message flows or message sets have been deployed to the execution group.

The basic configuration is now complete and the broker is ready to have message flow application resources deployed to it: 5.2, "Deploying message flow applications" on page 147 provides instructions on the methods of deploying these resources to a broker, and 5.3, "Publish/subscribe" on page 154 discusses topology deployment and publish/subscribe.

## 5.2 Deploying message flow applications

Deployment sends message flows to execution groups within brokers. Two methods can be used to deploy resources in the Message Broker Toolkit:

- Rapid Application Development deploy
- Deploy using a message broker archive

An additional method of deploying resources uses the mqsideploy command-line utility, but this is beyond the scope of this book.

To deploy message flow applications to an execution group, a Configuration Manager and broker must be running and must be part of the same domain.

### 5.2.1 Rapid Application Development deploy

Rapid Application Development (RAD) deploy is designed to be used in situations where message flow applications are being developed and changes are required frequently. With RAD, configuration files are created with properties for the execution group and broker to deploy to, as well as the resources to deploy, which means that the same resources can be deployed relatively rapidly to a broker and execution group after the initial configuration has been set up.

The “Preparing samples for first use” cheat sheet shown on page 69 uses RAD deploy to deploy sample message flows and message sets, so the RAD deploy wizard in 3.5, “Creating a default configuration” on page 55 works the same way. The Tasks view for the project that contains the message flow application resource cannot deploy if it displays error messages. Any errors that are seen in the Tasks view indicate that the message flows or message sets cannot be compiled without errors, such as syntax errors in the ESQL.

### 5.2.2 Setting up a RAD deploy configuration

To deploy any message flow application resource that exists in the workspace with RAD deploy:

1. Right-click a message flow or message set that you wish to deploy, and select **Run on Server** from the context menu.
2. Ensure that **Create a new server** is selected in the RAD deploy wizard that is displayed.
3. Click **Next**.
4. Ensure that the **Use an existing Configuration Manager connection file** option is selected.
5. Click **Next.**

At this point, the wizard connects to the Configuration Manager using the settings in the connection file, and the next panel of the wizard displays existing brokers and execution groups. The wizard does not allow a RAD deploy to an existing execution group, and instead generates a name for a new execution group based on the current user. However, the execution group name can be altered as long as the new name is not already being used as an execution group.

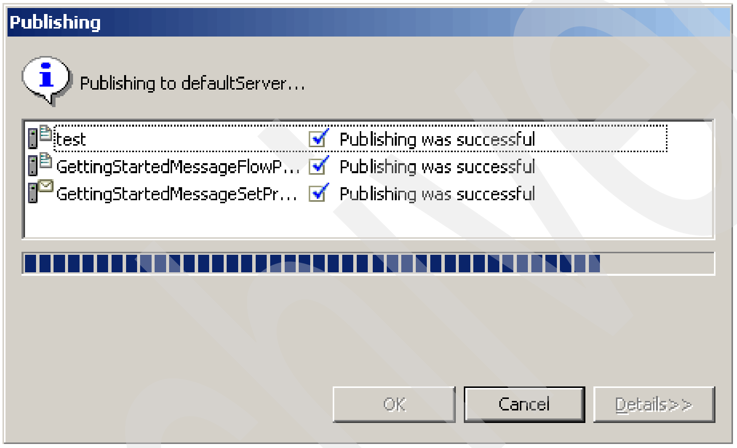
6. (Optional) Specify a new name in the Execution Group field if you do not wish to use the name generated by the wizard.
7. Click **Next**.
8. Select project names in the left-side panes to see the resources that can be deployed, and check the check boxes of the resources you wish to deploy.

9. Click **Finish**.

The dialog box shown in Figure 5-10 opens, displaying confirmation messages for each of the resources that are deployed to the execution group in the RAD deploy. This message may be seen each time a new RAD deploy configuration is set up and deployed to for the first time:

```
Error received while starting the server. Reason: Cannot start the server
because it is already running.
```

This message is not usually seen the next time the same RAD deploy configuration is used. It is normal behavior, and the deploy usually proceeds without any problems.


Publishing

Publishing to defaultServer...

test ✓ Publishing was successful
GettingStartedMessageFlowP... ✓ Publishing was successful
GettingStartedMessageSetPr... ✓ Publishing was successful

*Figure 5-10 Publishing confirmation in RAD Deploy*

When message flows that are linked to associated message sets through Project References are selected in the wizard, the associated message set is also automatically selected for deploy. If a message set is not selected automatically, then check the Project References for the message set or message flow project in the Broker Application Development perspective (outside of the RAD deploy wizard) by using the following instructions:

1. Right-click the message flow or message set project and select **Properties** from the context menu.
2. Click **Project References** in the left-side pane.
3. Check the check box next to the project to be referenced.
4. Click **OK** to make the change.

The end result of the RAD deploy is that the requested resources are deployed to a new execution group on the chosen broker and that two new files have been created in a Server project in the workspace. One of the files contains the set-up information for the broker and execution group for the deploy (for example, defaultServer.execgrp) and the other file contains the details of the message flow application resources to deploy (such as defaultConfiguration.execgrpcfg).

Check for the following success messages in the Event Log in the Message Brokers Toolkit to verify that the deploy operation was successful: BIP2056 and BIP4040.

### 5.2.3 Using an existing RAD deploy configuration

After a RAD deploy configuration has been set up as demonstrated in the previous section, the same resources can be deployed rapidly after changes have been made (for example, during initial message flow application development).

To use a RAD deploy with an existing configuration:

1. Right-click a message flow or message set file.
2. Ensure that **Use an existing server** is selected.
3. Choose the required server configuration (for example, defaultServer).
4. Click **Finish**.

This action starts the RAD deploy with the Publishing dialog box shown in Figure 5-10 on page 149. Check for the following success messages in the Event Log in the Message Brokers Toolkit to verify that the deploy operation was successful: BIP2056 and BIP4040.

### 5.2.4 Deploy using a message broker archive

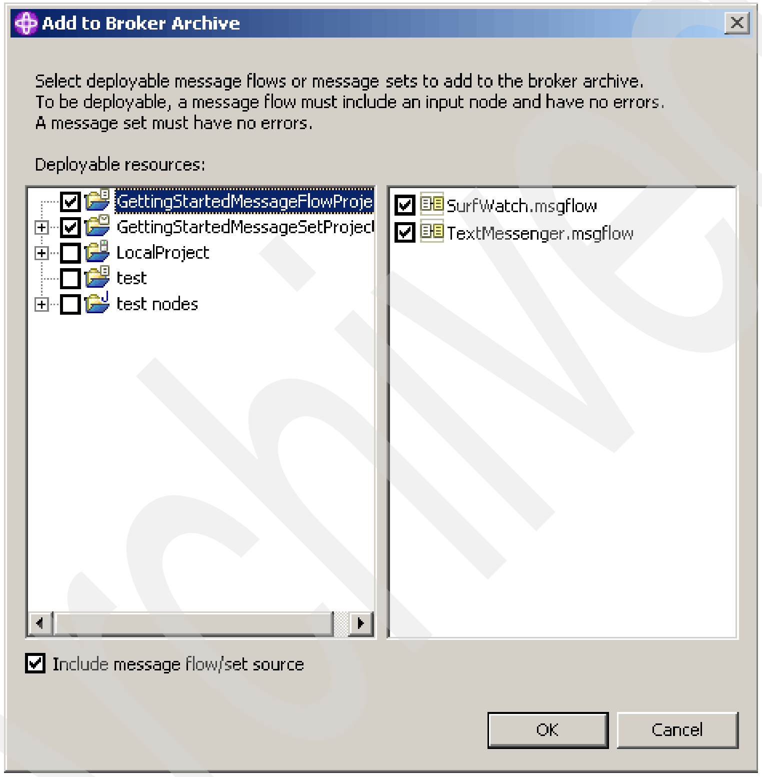
The second method of deploying message flow applications through the Message Brokers Toolkit uses a message broker archive to deploy the resources. A message broker archive is an archive file that contains any number of compiled message sets and message flows for deploy to an execution group in a broker. A broker archive can also contain source files. Message broker archives can be used as a mechanism for moving deployables between machines and backing up these important deployable resources. Message broker archives also can be used to change properties relating to the message flow. These are known as the *configurable* properties in the message broker archive.

### 5.2.5 Creating a message broker archive

The only requirement for creating a message broker archive is the availability of error-free message flow application resources. (Deploying a message broker archive, however, requires at least a simple configuration composed of a

Configuration Manager and broker.) To create a new Message Broker Archive in the Broker Administration perspective of the Message Brokers Toolkit:

1. Select **New** → **Message Broker Archive** from the **File** menu.
2. This opens a message broker archive editor. Click **Add** in the button bar.
3. In the “Add to broker archive” wizard, check the check boxes for the resources you wish to deploy, and check the check box at the bottom-left of the wizard to include message application source files (Figure 5-11).



Select deployable message flows or message sets to add to the broker archive.
To be deployable, a message flow must include an input node and have no errors.
A message set must have no errors.

- [✓] GettingStartedMessageFlowProje
- [✓] GettingStartedMessageSetProject
- [ ] LocalProject
- [ ] test
- [ ] test nodes

- [✓] SurfWatch.msgflow
- [✓] TextMessenger.msgflow

[✓] Include message flow/set source

*Figure 5-11 Add resources to a broker archive file*

Message flows and message sets can be added to the Message Broker Archive only if there are no errors in their containing message flow or message set project. Also, the message flows or sets must be in the correct location within a valid message flow or message set project.

4. Click **OK** to add the selected resources to the message broker archive file.
5. The selected message flow application resources are compiled and a dialog box opens. Click **Details** to view the outcome of the operation.

6. Click **OK**. The successfully added resources are displayed in the message broker archive editor.
7. Check the check box at the bottom-left of the message broker archive editor to display or hide the source files if any in the displayed resources.
8. Save the message broker archive by clicking **Save** in the button bar.

## 5.2.6 Deploying a message broker archive

There are two ways to deploy a message broker archive within the Message Brokers Toolkit:

- Drag-and-drop
- Deploy File

Both require an established connection to the Configuration Manager.

### Deploying a message broker archive using drag-and-drop

This is the simplest method of deploying a message broker archive, but care must be taken to avoid inadvertently dropping the message broker archive into the wrong execution group. (This can be corrected in the Domains view by right-clicking the deployed resource and choosing Delete from the context menu.)

To perform a drag-and-drop deploy:

1. In the Broker Administration perspective, expand the **Broker Topology** in the Domains view to see the available brokers and execution groups.
2. In the Broker Administration Navigator view, select the message broker archive (.bar) file to deploy.
3. Hold the mouse button down over the message broker archive file and drag the file over the execution group where you wish it to be deployed.
4. Let the mouse button go over the execution group.

An indication of whether the file is over an execution group is given by the mouse pointer during dragging, as it changes shape to a circle with a diagonal line through it when the mouse is not positioned in the correct place for the deploy. At the point that the message broker archive file is dropped onto the execution group, the deploy has been initiated and a confirmation message is received from the Configuration Manager if the deploy operation was successful.

As with RAD deploy, check the Event Log in the Message Brokers Toolkit for the BIP2056 and BIP4040 success messages in the Event Log in the Message Brokers Toolkit to verify that the deploy operation was successful.

## Deploying a message broker archive using deploy file

This method is slightly more complex than the drag-and-drop method, as it involves initiating the deploy using a menu and a wizard, but it is less prone to accidental deployment to the wrong execution group.

To deploy a message broker archive using Deploy File:

1. Select the message broker archive (.bar) file to deploy in the Broker Administration Navigator view.
2. Right-click the .bar file and select **Deploy File** from the context menu.
3. This opens the Deploy a BAR File wizard, which shows brokers and their execution groups in the broker domain (Figure 5-13 on page 156).

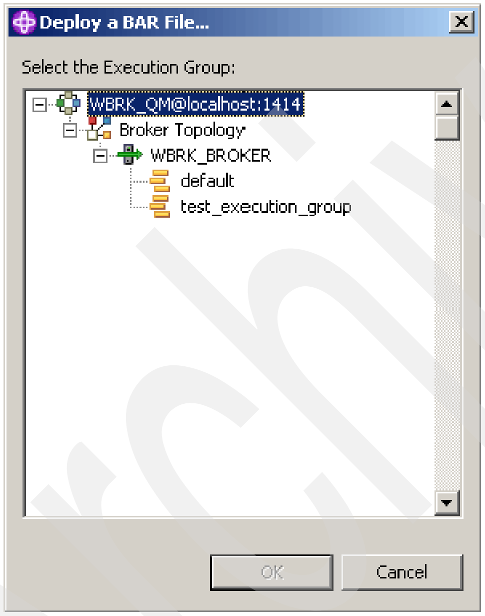


*Figure 5-12 .bar file deploy*

4. Select the target execution group, and click **OK** to perform the deployment.

A confirmation message is received from the Configuration Manager if the deploy operation was successful. Check the Event Log in the Message Brokers Toolkit for the BIP2056 and BIP4040 success messages to verify that the deploy operation was successful.

## 5.3 Publish/subscribe

Publish/subscribe is a style of messaging in which messages that are produced by a single application can be received and utilized by many other applications. Publish/subscribe is a broad topic, and an in-depth discussion of its uses and configuration is beyond the scope of this book. The Broker Administration perspective in the Message Brokers Toolkit includes a number of tools to assist in the configuration and administration of a publish/subscribe broker network, so a brief description of the function of each tool is given here with a short overview of publish/subscribe concepts.

As an additional source of information, and to try out some examples of publish/subscribe using the Message Brokers Toolkit, try the Soccer and Scribble samples that are provided in the samples gallery in the help system, as well as the SurfWatch Getting Started sample. Note that the Getting Started samples and Scribble sample are not supplied with WebSphere Business Integration Event Broker as they contain nodes that are not available on that product, and the Getting Started sample has a message set defined.

### Publish/subscribe basics

An application known as a *publisher* generates a message that can be utilized by other receiving applications known as *subscribers*. The published message contains publish/subscribe commands and information in the message header. A published message is called a *publication*. There are two basic types of publish/subscribe commands contained in the header: one to register a subscription and the other to publish a message. Both messages show a message topic. A subscriber subscribes to messages of a particular topic using a subscription message, and a publisher publishes a message about a given topic.

This is an example from the SurfWatch sample of a message for registering a subscription:

```
<psc><Command>RegSub</Command><Topic>SouthShore</Topic><Topic>SunsetBeach</
Topic><Topic>Rockpile</Topic><Topic>Kahaluu</Topic><RegOpt>PersAsPub</RegOp
t><QName>PAGER</QName></psc>
```

This is an example from the SurfWatch sample of a publication:

```
<psc><Command>Publish</Command><Topic>Laniakea</Topic></psc><mcd><Msd>mrm</
Msd><Set>GettingStartedMessageSets</Set><Type>Pager</Type><Fmt>XML</Fmt></m
cd><Pager><Text>SurfWatch12.02.0420:57:18Laniakea: Onshore, waves
Om.</Text></Pager>
```

The various tools in the Broker Administration perspective can be used to set up collections of brokers for use in publish/subscribe networks and to manage topics and subscriptions as described in the following sections.

## 5.3.1 Broker topology

Many brokers can be added to a broker domain and can be configured for publish/subscribe through the Broker Topology editor in the Message Brokers Toolkit. Brokers that are added to the broker domain are visible in the Broker Topology editor. Properties for individual brokers can be edited here; these are advanced properties relating to publish/subscribe security, multicast, and so on.

Also in the Broker Topology editor, brokers can be connected in a *collective*, which is a connecting of brokers that enables publish/subscribe messages to flow between them. In order to combine two brokers into a collective, the queue managers must be connected for communication using channel sender and receiver pairs/ This is in addition to any channels that connect the queue managers for the brokers to the Configuration Manager queue manager.

Setting up communication for multiple brokers is relatively straightforward, but remember the following rules:

- Create a sender channel on each queue manager with the same name as the receiver channel on the other queue manager.
- In the sender channel properties, make the Connection Name the same as the machine to connect to. Name the listener on the queue manager in this format: MachineName(listener port), for example BRK001(1728).
- Create a transmission queue with the same name as the queue manager to connect to, and set this as the Transmission queue in the sender channel.
- Confirm that the channel communication starts successfully.

Brokers can be added to the broker domain through the Broker Administration Perspective in the Domains view or through the Broker Topology editor.

To add a broker to the Broker Topology:

1. Double-click **Broker Topology** in the Domains view to open the Broker Topology editor.l
2. In the Broker Topology editor under **Entity**, click **Broker**.
3. Click on the canvas of the Broker Topology editor to add a broker to the Broker Topology.
4. Right-click the new broker and select **Properties** to set the Broker name and the Queue Manager name. Other properties for publish/subscribe and multicast can be set up here if required.

To connect two brokers in a collective:

1. In the palette pane of the Broker Topology editor, under **Entity**, select **Collective**.

2. Click on the canvas of the Broker Topology editor to add a collective to the Broker Topology.
3. Click the Connection tool in the palette part of the Broker Topology editor. Use the Connection tool to connect two brokers to the collective by creating two connections in the same way that connections between message flows nodes are made in the Message Flow Editor.
4. Save the editor contents (**File** → **Save**). The save option has the following format: `Save WBRK_QM Topology`.
5. A Topology Configuration Deploy message is displayed (Figure 5-8 on page 146). Select either **Delta** or **Complete** to deploy the configuration changes that have been made to the brokers.

Figure 5-13 shows a collective of two brokers: WBRK_BROKER and WBRK_BROKER2.

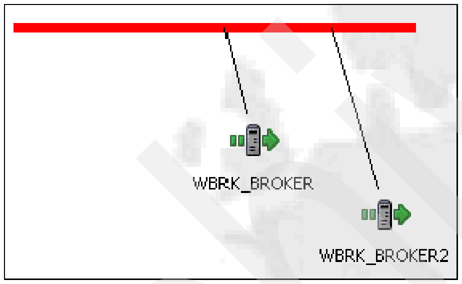


*Figure 5-13 Brokers joined in a collective*

Using a collective enables publish/subscribe messages to be communicated across brokers.

## 5.3.2 Topics

Topics that are defined in publication messages and in subscription messages control the routing of publish/subscribe messages. When a publication is created by an application, a message that passes through a publish/subscribe message flow is routed to applications that have subscribed to receive messages on that topic. You can set up definitions of topics and subtopics within the Message Brokers Toolkit, and these can be used to configure topic-based security to limit which applications can access which messages passing through the system.

In order to view users and groups for setting security on topics, the following tasks must have been performed:

- A User Name Server must be been set up (ideally using the same queue manager as the Configuration Manager).

- Channels between the User Name Server queue manager and any brokers queue managers must be set up.
- Use the **mqsichangebroker** command to set the -j parameter, and to set the queue manager for the User Name Server onto any brokers in the domain, using the following format:

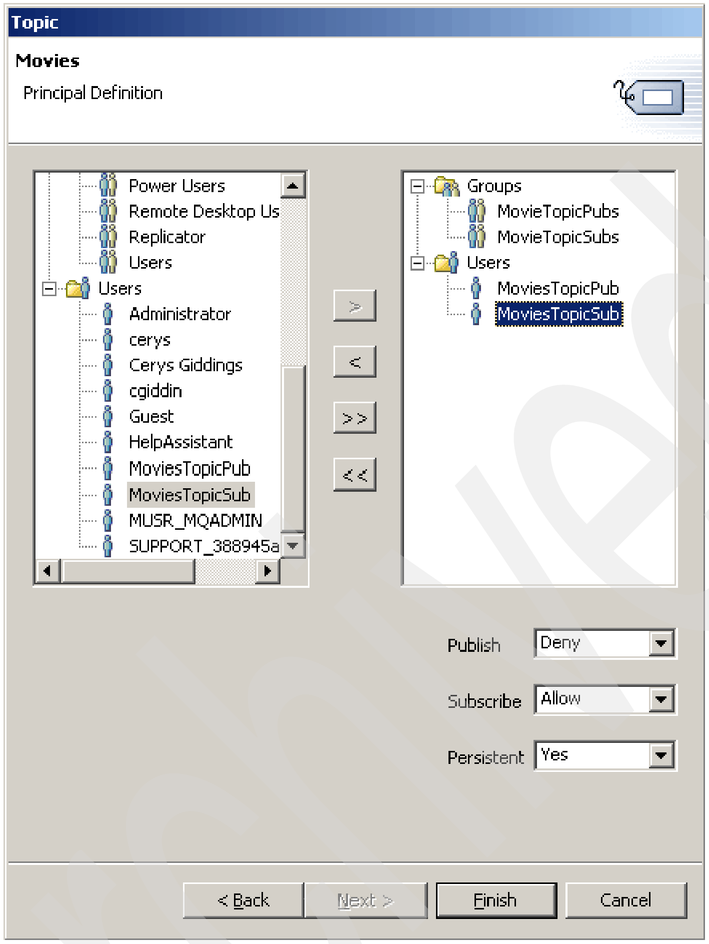
```
mqsichangebroker WBRK_BROKER -j -s WBRK_QM
```

- Use the **mqsichangeconfigmgr** command to set the User Name Server queue manager on the Configuration Manager using the following format:

```
mqsichangeconfigmgr -s WBRK_QM
```

These instructions give an overview for creating topics and assigning security to assign the publishers and subscribers who can use the topics that are created:

1. In the Broker Administration perspective, double-click **Topics** in the Domains view.
2. In the Topics Hierarchy editor that is displayed, right-click in the Topics section and select **Create Topic** from the context menu.
3. This opens the Topic wizard. Enter a name for the topic.
4. Click **Next**.
5. On the **Principal Definition** page (Figure 5-15 on page 159), define the security settings for users and groups using the options to **Deny** or **Allow** subscription or publication on this new topic. These can be added to and edited later.
6. Click **Finish** to complete the task.


Topic

**Movies**

Principal Definition

Power Users
Remote Desktop Us
Replicator
Users
Users
Administrator
cerys
Cerys Giddings
cgiddin
Guest
HelpAssistant
MoviesTopicPub
MoviesTopicSub
MUSR_MQADMIN
SUPPORT_388945a

Groups
MovieTopicPubs
MovieTopicSubs
Users
MoviesTopicPub
MoviesTopicSub

Publish: Deny
Subscribe: Allow
Persistent: Yes

*Figure 5-14 Principal Definition in Topic Wizard*

Other topics and subtopics can be created in the same way. To create a subtopic, select the named topic rather than the Topic root on the first page of the Topic wizard.

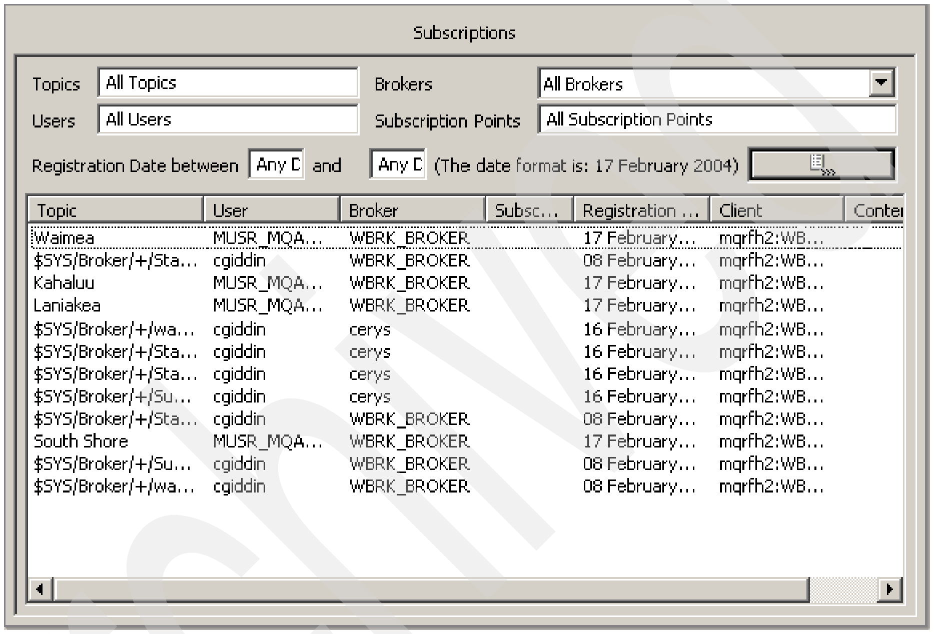
## 5.3.3 Subscriptions

When a subscriber application sends a subscription request to a broker on a particular topic, the broker stores the subscription data in a Subscription Table. The data in this Subscription Table can be accessed and searched using the Message Brokers Toolkit in the Subscriptions Query editor.

To show all subscriptions that are registered in the Subscription Table:

1. In the Broker Administration perspective, double-click **Subscriptions** in the Domains view.
2. In the Subscriptions Query editor, click the **Query** button.

This displays all of the Subscriptions that are registered with all of the brokers that are set up for publish/subscribe, as shown in Figure 5-15.



Topics: All Topics
Brokers: All Brokers
Users: All Users
Subscription Points: All Subscription Points
Registration Date between: Any D and Any D (The date format is: 17 February 2004)

| Topic | User | Broker | Subsc... | Registration ... | Client | Conte |
|---|---|---|---|---|---|---|
| Waimea | MUSR_MQA... | WBRK_BROKER | | 17 February... | mqrfh2:WB... | |
| $SYS/Broker/+/Sta... | cgiddin | WBRK_BROKER | | 08 February... | mqrfh2:WB... | |
| Kahaluu | MUSR_MQA... | WBRK_BROKER | | 17 February... | mqrfh2:WB... | |
| Laniakea | MUSR_MQA... | WBRK_BROKER | | 17 February... | mqrfh2:WB... | |
| $SYS/Broker/+/wa... | cgiddin | cerys | | 16 February... | mqrfh2:WB... | |
| $SYS/Broker/+/Sta... | cgiddin | cerys | | 16 February... | mqrfh2:WB... | |
| $SYS/Broker/+/Sta... | cgiddin | cerys | | 16 February... | mqrfh2:WB... | |
| $SYS/Broker/+/Su... | cgiddin | cerys | | 16 February... | mqrfh2:WB... | |
| $SYS/Broker/+/Sta... | cgiddin | WBRK_BROKER | | 08 February... | mqrfh2:WB... | |
| South Shore | MUSR_MQA... | WBRK_BROKER | | 17 February... | mqrfh2:WB... | |
| $SYS/Broker/+/Su... | cgiddin | WBRK_BROKER | | 08 February... | mqrfh2:WB... | |
| $SYS/Broker/+/wa... | cgiddin | WBRK_BROKER | | 08 February... | mqrfh2:WB... | |

*Figure 5-15 Subscription Query Editor*

These registered subscriptions can be queried on broker, topic, users, dates, and subscription points. Subscriptions do not remain in the Subscription Table indefinitely, as they can be de-registered by the subscriber, expired, or deleted. They are also removed from the table when a subscriber application stops.

The Subscription Query editor can be used to delete subscriptions from brokers:

1. Right-click on the subscription you want to delete and select **Delete** from the context menu.
2. Click **OK** in the confirmation dialog box.

A configuration change is sent to the broker via the Configuration Manager to remove the subscription from the Subscription Table. The Event Log in the Broker Administration perspective can be checked for success or failure messages.

6

# Troubleshooting and problem determination

This chapter provides assistance with determining the cause and resolution of problems when using WebSphere Business Integration Message Brokers. The following topics are discussed in this chapter:

- Locating error information
- Message flow debugger
- Using trace
- Troubleshooting common problems
- Other advice

## 6.1 Locating error information

This section details the location where errors generated by WebSphere Business Integration Message Brokers may be found. There are many locations with information that is useful for problem determination; some are within the Message Brokers Toolkit, but others must be located in the directory structure of the system that is running the product.

### 6.1.1 Error messages

Error messages are usually the first indication that a problem may exist and must be resolved in some part of the WebSphere Business Integration Message Brokers configuration or application. These messages are displayed in the Message Brokers Toolkit or on the command line in response to a command or a command line utility. Information about the different types of messages that may be seen in the Message Brokers Toolkit and on the command line are detailed here with guidance for interpreting them.

#### Message structure

Messages generated by the WebSphere Business Integration Message Brokers product have the following structure:

```
BIP8081E
```

- `BIP` is the three-character product family identity, the same as in previous versions of the product.
- The four-character number following BIP is the unique identifier of the message. This indicates the condition that generated the message in the product and can be looked up in the product help or message catalog for further information. (Details of how to find more information about messages is given later in this section.)
- The final character indicates the message type: `I` for an information message, `W` for a warning message, or `E` for an error message.

The BIP message code is usually accompanied by a description of the condition that generated the message, and sometimes a reason or a suggestion.

#### Information, warnings, and errors

Messages generated by WebSphere Business Integration Message Brokers may be classified as one of three types:

- Information
- Warning
- Error

Information messages simply provide information to the user and do not represent any problem of concern. Success conditions are classed as information messages; for example, BIP8096I is the message for successful command initiation on the command line, and BIP0892I is the message for a successful response being received from the Configuration Manager in the Message Brokers Toolkit as in Figure 6-1 on page 164).

Warning messages are less severe than error messages and do not represent an immediate problem, but they indicate situations that may need investigation. These messages are less likely to be seen on the command line or as pop-ups in the Message Brokers Toolkit, and are more likely seen in the system log of the machine where the warning occurred. Warning messages without a BIP code are generated by the Message Brokers Toolkit and can be seen in the Tasks View in the Broker Application Development Perspective and the Alerts View in the Broker Administration Perspective of the Message Brokers Toolkit.

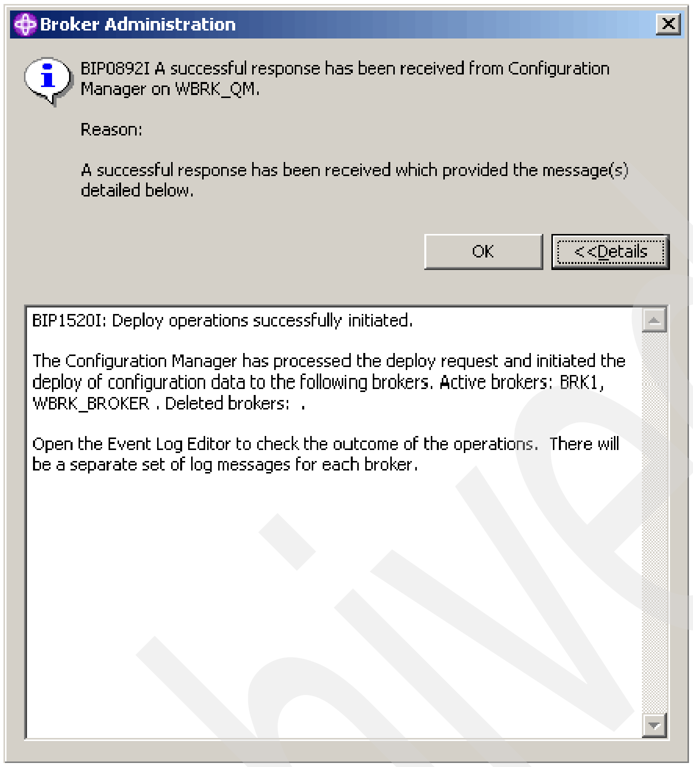
The most severe and important messages are the error messages that are generated by the WebSphere Business Integration Message Brokers product. These can be seen on the command line, in the Message Brokers Toolkit, and in the system log of the machine where the product is running. These messages are either generated immediately in response to a failed action, such as trying to start a broker that does not exist, or as a response to a failure during the running of the product. Messages seen on the command line or as pop-ups in the Message Brokers Toolkit are failure responses to an action from the user.

Messages that are found in the system log or in the Message Brokers Toolkit Event Log usually occur during the running of the product, such as a message that is produced when a message flow attempts to access a WebSphere MQ queue that does not exist. These errors do not occur as a result of user action.

## Messages within the Message Broker Toolkit

Messages displayed within the Message Broker Toolkit are of two basic types: pop-up messages that are generated as instant feedback to an action, and messages that result from the running of the product and are displayed in a log or a view, such as the results of a deploy operation in the Event Log or an error produced when a message flow containing an error is saved.

An example of a pop-up message is shown in Figure 6-1 on page 164, which shows a success message that was received from the Configuration Manager after a deploy action. This figure also shows the typical structure of a BIP message with the BIP code followed by a description and then a Reason section that details why the message is being displayed. There is also a Details section that can be made visible or hidden by clicking the **Details** button. This gives more detailed information about the message, and in an error message this contains suggestions about how to correct the problem.



BIP0892I A successful response has been received from Configuration Manager on WBRK_QM.

Reason:

A successful response has been received which provided the message(s) detailed below.

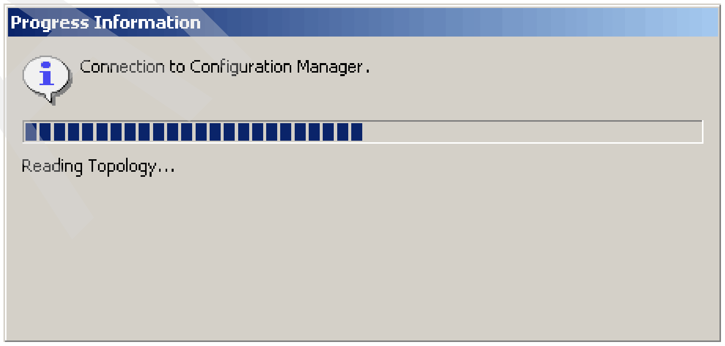
BIP1520I: Deploy operations successfully initiated.

The Configuration Manager has processed the deploy request and initiated the deploy of configuration data to the following brokers. Active brokers: BRK1, WBRK_BROKER . Deleted brokers:  .

Open the Event Log Editor to check the outcome of the operations.  There will be a separate set of log messages for each broker.

*Figure 6-1 Pop-up message from the Message Brokers Toolkit*

Pop-up messages are also displayed in the Message Brokers Toolkit. These are not BIP messages, and frequently are some kind of progress or status indicator, such as when messages have been added to a broker archive file. Figure 6-2 shows a progress message that is displayed when the Message Brokers Toolkit connects to the Configuration Manager.

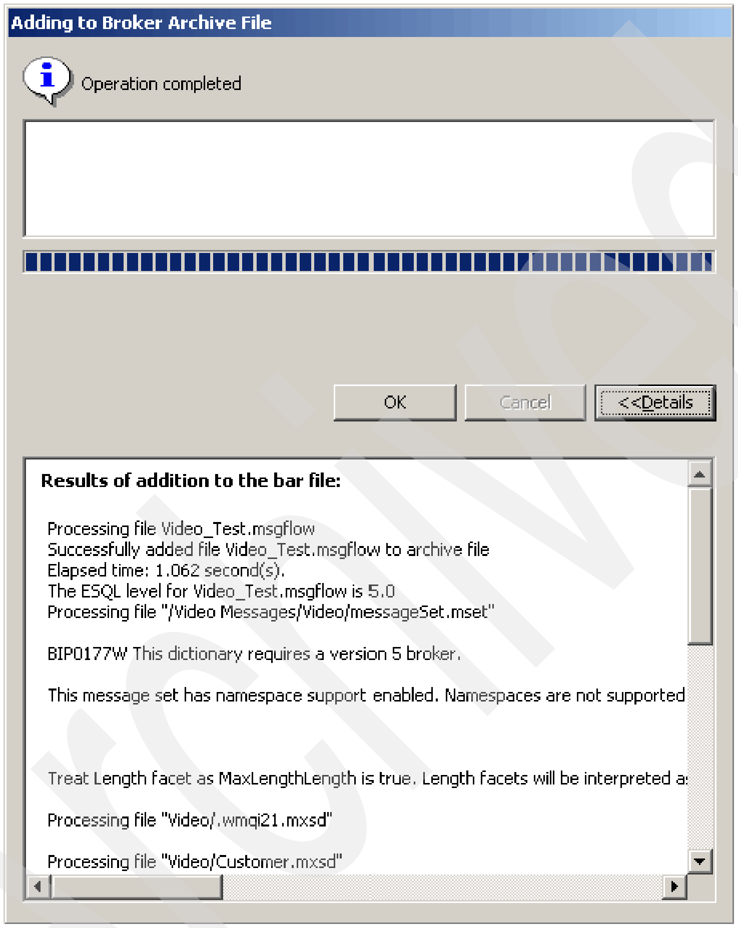


Connection to Configuration Manager.

Reading Topology...

*Figure 6-2 Progress message*

Figure 6-3 shows a status message. The Details section shows the status after the addition of some message flows and message sets to a broker archive file. If any problems occurred while adding the resources, those errors or warnings are displayed here.



Results of addition to the bar file:

```
Processing file Video_Test.msgflow
Successfully added file Video_Test.msgflow to archive file
Elapsed time: 1.062 second(s).
The ESQL level for Video_Test.msgflow is 5.0
Processing file "/Video Messages/Video/messageSet.mset"

BIP0177W This dictionary requires a version 5 broker.

This message set has namespace support enabled. Namespaces are not supported


Treat Length facet as MaxLengthLength is true. Length facets will be interpreted a:

Processing file "Video/.wmqi21.mxsd"

Processing file "Video/Customer.mxsd"
```

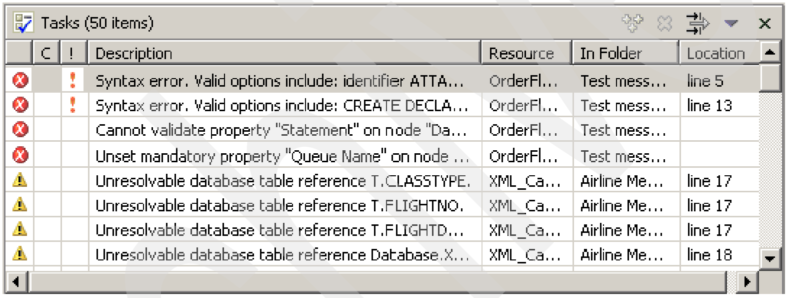
*Figure 6-3 Adding resources to the broker archive file*

In the figure, the message set that was added to the broker archive file contains namespaces. This prompts the warning message BIP0177W to be displayed to indicate that the message set is incompatible with brokers that are running on previous versions of the product. This is a good example of the difference between errors and warnings. In this case the message set is valid so there is no error, but if the message set is deployed to a WebSphere MQ Integrator Version 2.1 broker, then errors occur.

In addition to these pop-up messages, there are four other locations where messages are generated within the Message Brokers Toolkit. The Event Log editor and the Eclipse error log are discussed in upcoming sections. The other locations are in the Tasks view in the Broker Application Development perspective and in the Alerts view in the Broker Administration perspective.

The Tasks view in the Broker Application Development perspective shows information, warning, and error messages relating to the broker application development resources such as message definition files, ESQL files, mapping files, and message flows. Whenever one of these resources is saved, the resource is built and the contents are checked for errors. Any problems or warnings that are found are displayed in the Tasks View.

The information that is associated with these tasks is a symbol for the severity level: Information, Warning, and Error as indicated by the icons. There are also priority icons so that the most significant errors can be highlighted. Figure 6-4 shows an example of messages displayed in the Tasks view.



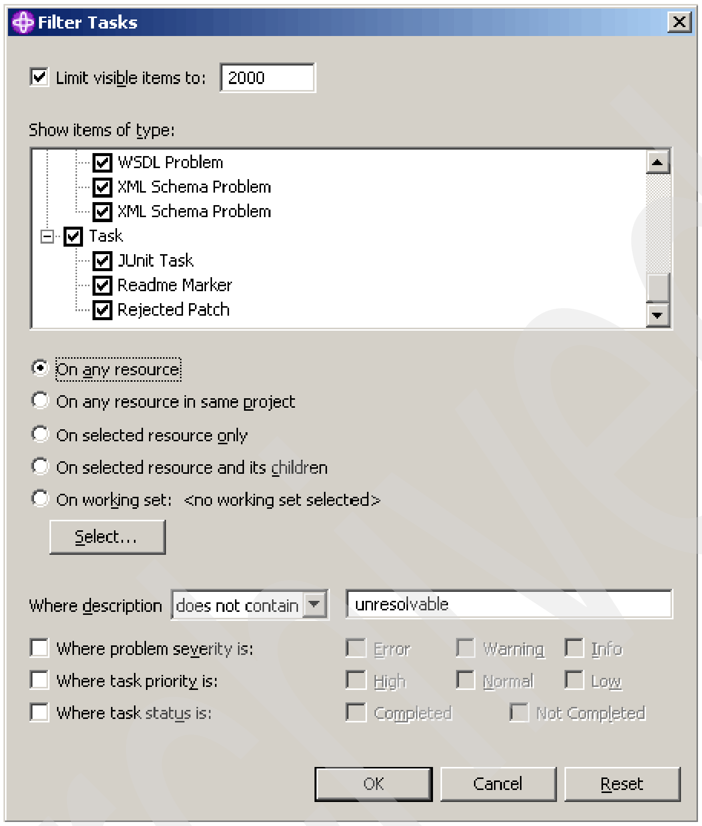
|  | C | ! | Description | Resource | In Folder | Location |
|---|---|---|---|---|---|---|
| ⊗ |  | ! | Syntax error. Valid options include: identifier ATTA... | OrderFl... | Test mess... | line 5 |
| ⊗ |  | ! | Syntax error. Valid options include: CREATE DECLA... | OrderFl... | Test mess... | line 13 |
| ⊗ |  |  | Cannot validate property "Statement" on node "Da... | OrderFl... | Test mess... |  |
| ⊗ |  |  | Unset mandatory property "Queue Name" on node ... | OrderFl... | Test mess... |  |
| ⚠ |  |  | Unresolvable database table reference T.CLASSTYPE. | XML_Ca... | Airline Me... | line 17 |
| ⚠ |  |  | Unresolvable database table reference T.FLIGHTNO. | XML_Ca... | Airline Me... | line 17 |
| ⚠ |  |  | Unresolvable database table reference T.FLIGHTD... | XML_Ca... | Airline Me... | line 17 |
| ⚠ |  |  | Unresolvable database table reference Database.X... | XML_Ca... | Airline Me... | line 18 |

*Figure 6-4 Messages in the Tasks view*

The top four messages that are displayed in the figure are error messages. The top two are marked as high priority, as these are syntax errors in ESQL. The rest of the visible messages are warning messages that relate to unresolvable database table references. These do not cause any problems with deploying the message flows they belong to, but if the database tables required by the message flow do not exist, then other errors are generated later by the runtime.

These messages can be sorted by clicking the column headings. For example, to sort these messages by severity, click the column heading where the severity icons are displayed (left-most column). To sort by resource, click the column heading labeled Resource.

Messages can be filtered using the Filter Tasks wizard. Click the Filter button (the button with three arrows in Figure 6-4) to launch this wizard. The example in Figure 6-5 on page 167 shows how to hide all messages containing the word *unresolvable*. The Filter wizard can be especially useful when a lot of similar messages are displayed in the Tasks view.


Filter Tasks

[✓] Limit visible items to: 2000

Show items of type:
- [✓] WSDL Problem
- [✓] XML Schema Problem
- [✓] XML Schema Problem
- [✓] Task
  - [✓] JUnit Task
  - [✓] Readme Marker
  - [✓] Rejected Patch

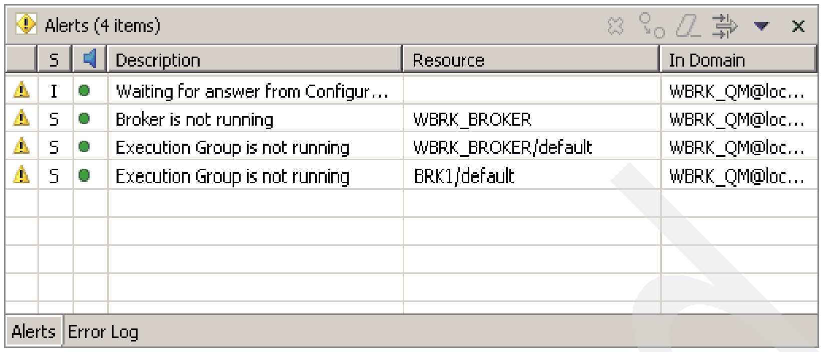
[✓ On any resource]
On any resource in same project
On selected resource only
On selected resource and its children
On working set: <no working set selected>

Where description does not contain unresolvable

[ ] Where problem severity is: Error Warning Info
[ ] Where task priority is: High Normal Low
[ ] Where task status is: Completed Not Completed

*Figure 6-5 Filter Tasks wizard*

The Alerts view in the Broker Administration perspective looks similar to the Tasks view in the way that it displays messages. However, the messages that it displays are the status of runtime components within a broker domain. This information is refreshed regularly so that if connections to the brokers in the domain are running correctly, then any changes to the brokers or execution group states are displayed. Figure 6-6 on page 168 shows an example of the messages that are displayed in the Alerts view.



Alerts (4 items)

| | S | | Description | Resource | In Domain |
|---|---|---|---|---|---|
| ⚠ | I | ● | Waiting for answer from Configur… | | WBRK_QM@loc… |
| ⚠ | S | ● | Broker is not running | WBRK_BROKER | WBRK_QM@loc… |
| ⚠ | S | ● | Execution Group is not running | WBRK_BROKER/default | WBRK_QM@loc… |
| ⚠ | S | ● | Execution Group is not running | BRK1/default | WBRK_QM@loc… |

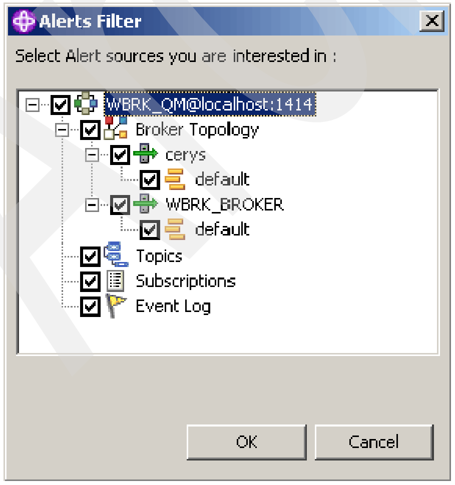
Alerts | Error Log

*Figure 6-6 Runtime status messages in the Alerts view*

The Alerts view shows that a deploy operation is in progress (Waiting for an answer from Configuration Manager), a broker called `WBRK_BROKER` has been stopped, and two execution groups are running. The running components do not have to be displayed here as they are displayed in the Domains view.

The Alerts cannot be sorted in the same way as the Tasks in the Message Application Development perspective, but they can be filtered. The Alerts Filter wizard enables the Alert sources to be selected. Alert sources such as individual brokers and execution groups can be deselected so that no alerts relating to these are displayed in the Alerts view.

The Alerts Filter wizard can be displayed by clicking on the Filter button (third from the left above the Alerts view shown in Figure 6-6). The layout of the Alerts Filter wizard can be seen in Figure 6-7.



Alerts Filter

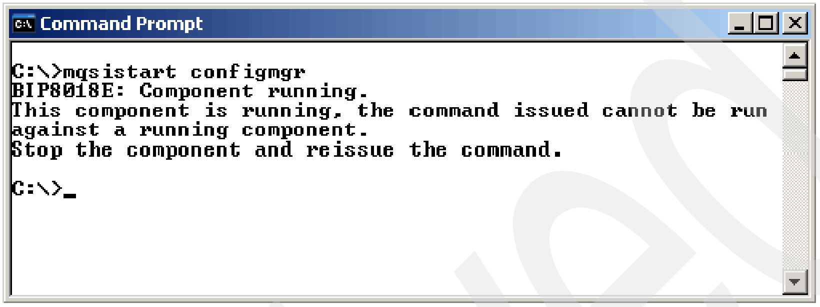
Select Alert sources you are interested in :

```
[✓] WBRK_QM@localhost:1414
    [✓] Broker Topology
        [✓] cerys
            [✓] default
        [✓] WBRK_BROKER
            [✓] default
    [✓] Topics
    [✓] Subscriptions
    [✓] Event Log
```

*Figure 6-7 Filtering Alerts*

## Messages on the command line

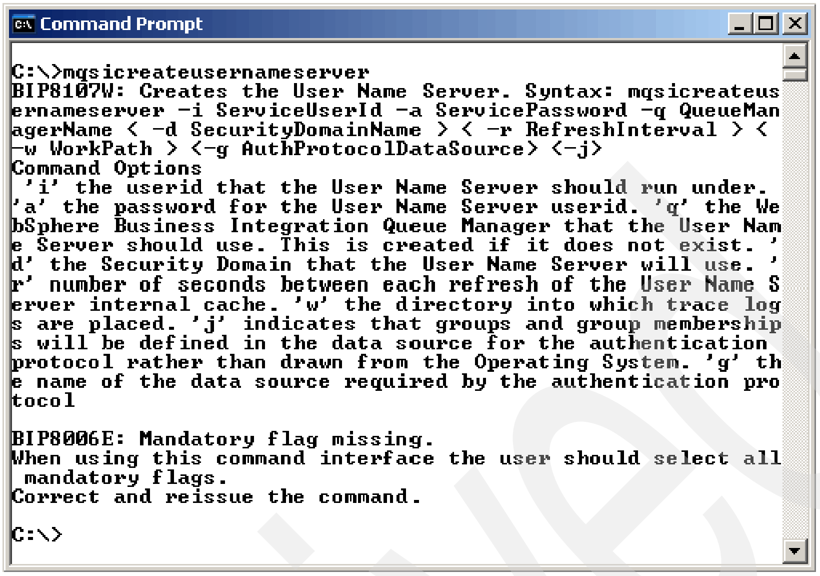
Messages are displayed on the command line in the same format as the BIP messages are displayed in the Message Broker Toolkit when commands have been run. Figure 6-8 shows an example of a typical command line message. The BIP code is displayed, followed by the description in the first sentence. The rest of the message gives the reason and a suggestion for solving the problem.



```
C:\>mqsistart configmgr
BIP8018E: Component running.
This component is running, the command issued cannot be run
against a running component.
Stop the component and reissue the command.

C:\>_
```

*Figure 6-8 A BIP message displayed on the command line*

For many of the WebSphere Business Integration Message Brokers commands, syntax assistance is displayed in addition to the message if the parameters that are used in the command are incorrect. This is a useful way to determine the syntax that must be used with these commands. For example, typing **mqsicreateusernameserver** on a command line results in the response seen in Figure 6-9 on page 170. This shows the syntax of the command and specific details as to the meanings or requirements of the parameters, and these are followed by the BIP8006E error message to indicate that the mandatory flags are missing.



```
C:\>mqsicreateusernameserver
BIP8107W: Creates the User Name Server. Syntax: mqsicreateus
ernameserver -i ServiceUserId -a ServicePassword -q QueueMan
agerName < -d SecurityDomainName > < -r RefreshInterval > <
-w WorkPath > <-g AuthProtocolDataSource> <-j>
Command Options
 'i' the userid that the User Name Server should run under.
'a' the password for the User Name Server userid. 'q' the We
bSphere Business Integration Queue Manager that the User Nam
e Server should use. This is created if it does not exist. '
d' the Security Domain that the User Name Server will use. '
r' number of seconds between each refresh of the User Name S
erver internal cache. 'w' the directory into which trace log
s are placed. 'j' indicates that groups and group membership
s will be defined in the data source for the authentication
protocol rather than drawn from the Operating System. 'g' th
e name of the data source required by the authentication pro
tocol

BIP8006E: Mandatory flag missing.
When using this command interface the user should select all
 mandatory flags.
Correct and reissue the command.

C:\>
```

*Figure 6-9 Help information generated by the mqsicreateusernameserver command*

## Locating more information about messages

In addition to the information that accompanies error messages, two other locations can be used to look up error messages and their explanations. The first of these is in the help system in the Message Brokers Toolkit (or other source of product help). Use the following instructions to locate information on a BIP message:

1. Open the Message Brokers Toolkit Help by selecting **Help** → **Help Contents**.
2. Click either **WebSphere Business Integration Message Broker** or **WebSphere Business Integration Event Broker**.
3. Select **Diagnostic messages** from the bottom of the Contents section of the Help page.
4. In the Diagnostic messages text box on the right side of the Help page, enter a four-character number and press Enter to search for a message with that number.
5. For each message, the following information is returned:
   - The BIP code
   - Severity
   - Message
   - Explanation
   - Response

The second place to find more information about BIP messages is in an HTML file called messages.html. This file is found in the messages directory in the install path. (For a typical Windows install, C:\Program Files\IBM\WebSphere Business Integration Message Brokers\messages.) Several directories exist here for translated messages, and in each is a messages.html file.

The messages.html file contains a list of BIP messages in order, in a table, with links to further information at the top of the file. Locate the required BIP message and click the link to view the information about it. This information is the same as the Help system except for the addition of an Inserts table for each message, which details the variable information for that message. For example, Table 6-1 shows the Inserts information for message BIP2623. This enables the source of variable information in the error message to be determined. If the error message in this example was received, then it is a logical step to look up the MQ reason code to help determine the cause of the error.

*Table 6-1 Example inserts for a BIP message*

| Insert number | Description | Datatype |
|---|---|---|
| &1 | MQ queue manager name | CHARACTER |
| &2 | MQ queue name | CHARACTER |
| &3 | MQ return code | CHRACTER |
| &4 | MQ reason code | CHARACTER |

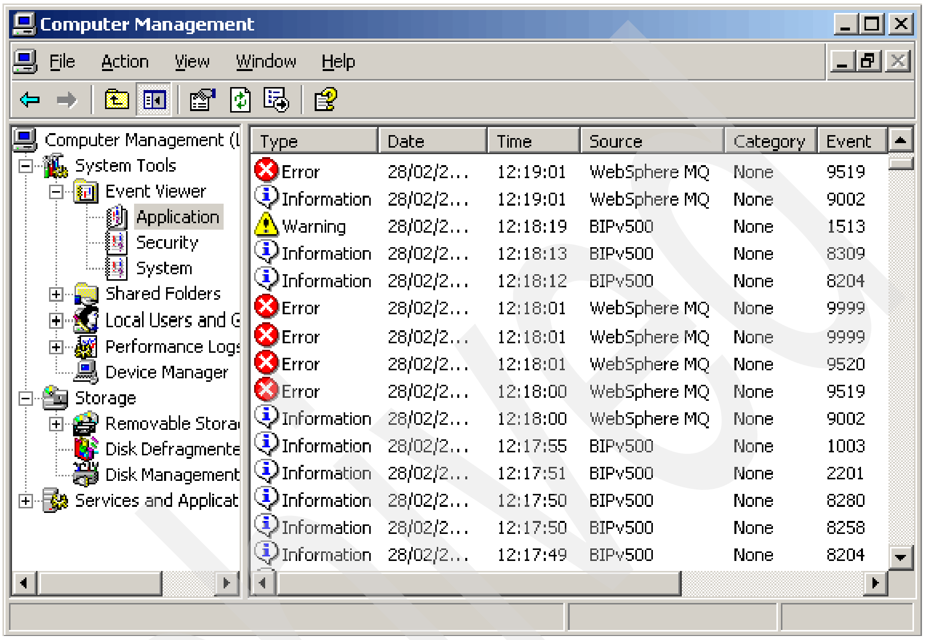
## 6.1.2 Windows Event Viewer

Messages that are generated by the WebSphere Business Integration Message Brokers runtime are recorded in the local error log. On Windows this is the Windows Event Log, on Unix it is the syslog, and on z/OS it is the operators console. These messages are all BIP messages that include information and warning messages as well as error messages. Where an error condition has occurred there may be multiple messages to describe the problem, generated by different components or parts of the runtime. In this section, we give information about locating and viewing messages using the Windows Event Log.

The exact location of the Windows Event Log may depend on the version of Windows that is installed, but typically it is found under Administrative Tools in the Windows Control Panel. It can also be accessed through the Computer Management option (right-click **My Computer** → **Manage** → **System Tools** → **Event Viewer**).

In the Windows Event Viewer, the BIP runtime messages from WebSphere Business Integration Message Brokers are recorded in the Application Log.

Figure 6-10 shows the Application Log through the Computer Management Tool in Windows. Messages that are produced by the runtime have a source of BIPv500, but messages generated by other programs such as WebSphere MQ and DB2 can also be found in the Application Log.



| Type | Date | Time | Source | Category | Event |
|---|---|---|---|---|---|
| Error | 28/02/2... | 12:19:01 | WebSphere MQ | None | 9519 |
| Information | 28/02/2... | 12:19:01 | WebSphere MQ | None | 9002 |
| Warning | 28/02/2... | 12:18:19 | BIPv500 | None | 1513 |
| Information | 28/02/2... | 12:18:13 | BIPv500 | None | 8309 |
| Information | 28/02/2... | 12:18:12 | BIPv500 | None | 8204 |
| Error | 28/02/2... | 12:18:01 | WebSphere MQ | None | 9999 |
| Error | 28/02/2... | 12:18:01 | WebSphere MQ | None | 9999 |
| Error | 28/02/2... | 12:18:01 | WebSphere MQ | None | 9520 |
| Error | 28/02/2... | 12:18:00 | WebSphere MQ | None | 9519 |
| Information | 28/02/2... | 12:18:00 | WebSphere MQ | None | 9002 |
| Information | 28/02/2... | 12:17:55 | BIPv500 | None | 1003 |
| Information | 28/02/2... | 12:17:51 | BIPv500 | None | 2201 |
| Information | 28/02/2... | 12:17:50 | BIPv500 | None | 8280 |
| Information | 28/02/2... | 12:17:50 | BIPv500 | None | 8258 |
| Information | 28/02/2... | 12:17:49 | BIPv500 | None | 8204 |

*Figure 6-10 Computer Management: Application Log*

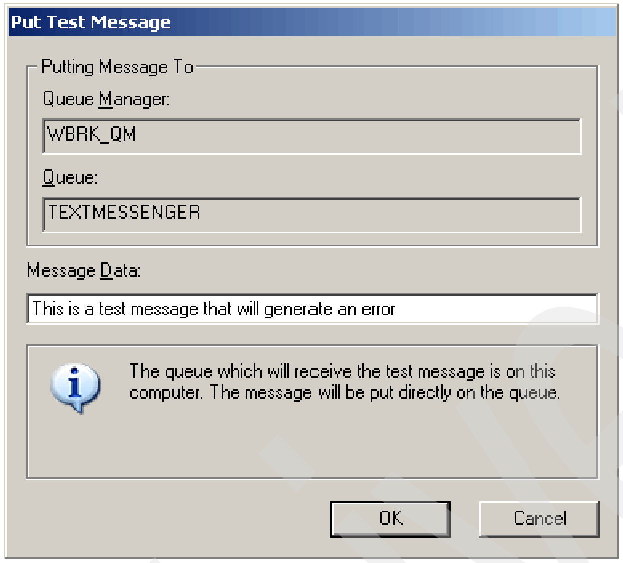
The BIP code can also be seen in this view in the Event column (for example, the BIP1513W message, which is a Warning type).

To view more details about any message in the Windows Event Viewer, double-click the message to open it in a window called Event Properties, where message details can be seen.

To navigate messages using the Windows Event Viewer by deliberately causing an error to occur in the runtime, follow these steps. (Putting an invalid message through the Getting Started Sample message flows can be used to generate an error condition in the runtime.)

1. Using the cheat sheet called Import the Getting Started Samples, create the required resources and deploy the sample message flows and message set.
2. Open the **WebSphere MQ Explorer** and navigate to the **TEXTMESSENGER** queue.
3. Right-click the queue and select **Put Test Message** from the context menu.

4. Enter a message to put to the queue on the Message Data line (Figure 6-11), and click **OK**.


Put Test Message

Putting Message To

Queue Manager: WBRK_QM

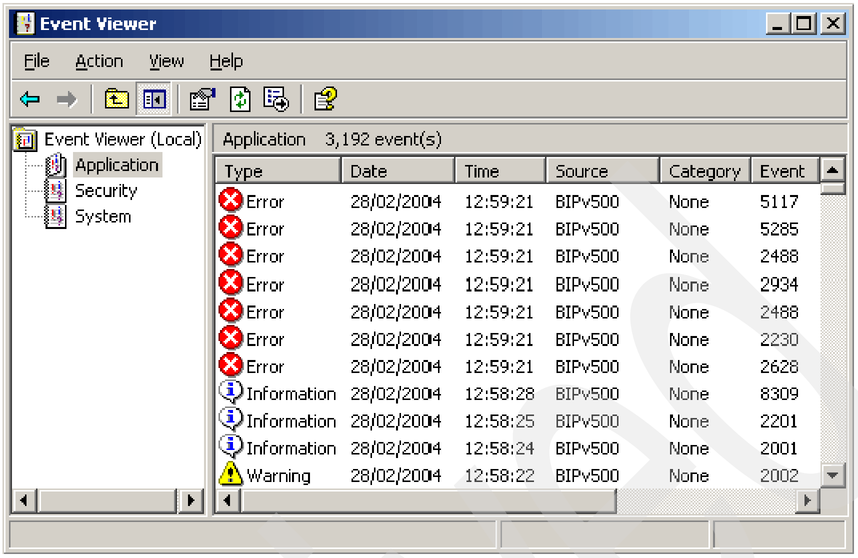
Queue: TEXTMESSENGER

Message Data: This is a test message that will generate an error

The queue which will receive the test message is on this computer. The message will be put directly on the queue.

*Figure 6-11 WebSphere MQ Explorer: Put Test Message utility*

5. Open the **Application Log** in the Windows Event Viewer and choose **Action** → **Refresh** to ensure that the latest messages are displayed. Figure 6-12 on page 174 shows the errors that are expected for an event of this type.
6. Double-click the first error message to display the details of the error. By default the events are sorted by date and time order, with the most recent message displayed at the top of the list. In this example, a BIP2628 is the first message to be displayed, with seven messages in all generated by the event.

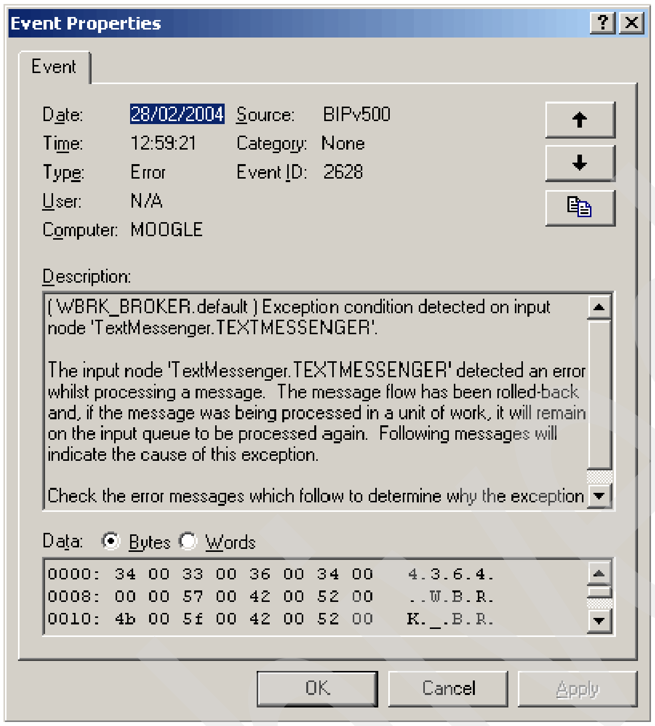


Application 3,192 event(s)

| Type | Date | Time | Source | Category | Event |
|---|---|---|---|---|---|
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 5117 |
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 5285 |
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 2488 |
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 2934 |
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 2488 |
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 2230 |
| Error | 28/02/2004 | 12:59:21 | BIPv500 | None | 2628 |
| Information | 28/02/2004 | 12:58:28 | BIPv500 | None | 8309 |
| Information | 28/02/2004 | 12:58:25 | BIPv500 | None | 2201 |
| Information | 28/02/2004 | 12:58:24 | BIPv500 | None | 2001 |
| Warning | 28/02/2004 | 12:58:22 | BIPv500 | None | 2002 |

*Figure 6-12 BIP error messages in the Application Log*

7. Figure 6-13 on page 175 shows an example of the BIP2628 message details: The message indicates that an exception has occurred on the input node in the TextMessenger message flow in the default execution group on broker WBRK_BROKER. The message has been rolled back to the input queue, and further error messages indicate the cause of the problem.



Date: 28/02/2004 Source: BIPv500
Time: 12:59:21 Category: None
Type: Error Event ID: 2628
User: N/A
Computer: MOOGLE

Description:

( WBRK_BROKER.default ) Exception condition detected on input node 'TextMessenger.TEXTMESSENGER'.

The input node 'TextMessenger.TEXTMESSENGER' detected an error whilst processing a message. The message flow has been rolled-back and, if the message was being processed in a unit of work, it will remain on the input queue to be processed again. Following messages will indicate the cause of this exception.

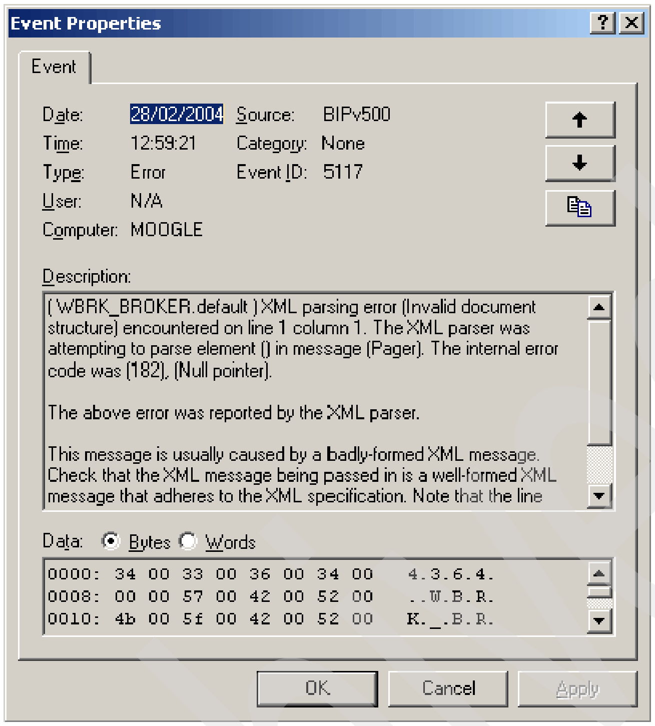
Check the error messages which follow to determine why the exception

Data: [✓ Bytes] Words

```
0000:  34 00 33 00 36 00 34 00   4.3.6.4.
0008:  00 00 57 00 42 00 52 00   ..W.B.R.
0010:  4b 00 5f 00 42 00 52 00   K._.B.R.
```

*Figure 6-13 An example BIP error message properties*

8. Click the up arrow at the top-right of the Event Properties window to view the next message in the sequence.
9. Read the rest of the messages to see what problems have occurred to prevent processing of the message. As with many similar error message groups, the final error message seen in the Application Log is the one that contains the original cause of the failure, and the other messages indicate the effect that the problem had on message processing. The final message is a BIP5117 message (Figure 6-14 on page 176), indicating that an XML parsing error occurred and that the message is badly formed XML. It also indicates that the message it was trying to parse was a Pager message.



Date: 28/02/2004 Source: BIPv500
Time: 12:59:21 Category: None
Type: Error Event ID: 5117
User: N/A
Computer: MOOGLE

Description:

( WBRK_BROKER.default ) XML parsing error (Invalid document structure) encountered on line 1 column 1. The XML parser was attempting to parse element () in message (Pager). The internal error code was (182). (Null pointer).

The above error was reported by the XML parser.

This message is usually caused by a badly-formed XML message. Check that the XML message being passed in is a well-formed XML message that adheres to the XML specification. Note that the line

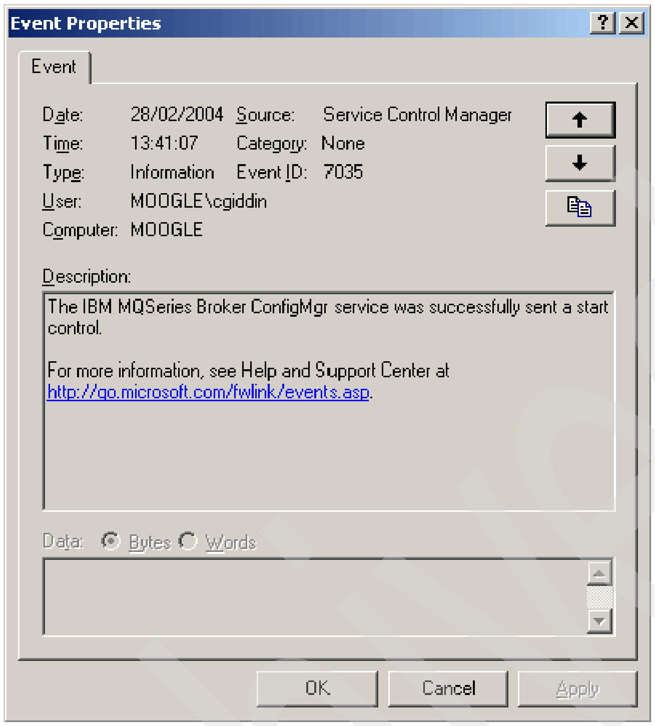
Data: [✓ Bytes] Words

```
0000:  34 00 33 00 36 00 34 00   4.3.6.4.
0008:  00 00 57 00 42 00 52 00   ..W.B.R.
0010:  4b 00 5f 00 42 00 52 00   K._.B.R.
```

*Figure 6-14 BIP error message showing a parser error*

The cause of the failing was that the message flow expected an XML message with the structure defined as a Pager message within the GettingStartedMessageSet definitions. The test message that was put through the WebSphere MQ Explorer does not match that definition, so the message cannot be processed and these error messages are recorded in the Windows Event Viewer Application Log.

Information about the starting, running, and stopping of WebSphere Business Integration Message Broker components can also be obtained from the System Log in the Windows Event Viewer. This is because the components such as brokers, the Configuration Manager, and User Name Server are created as Windows Services. When commands are sent to these components to start or stop them, then messages are written to the System Log to indicate the status of the command. Figure 6-15 on page 177 shows an example of a message from the System Log that indicates that a start message has been sent to the Configuration Manager.



Date: 28/02/2004 Source: Service Control Manager
Time: 13:41:07 Category: None
Type: Information Event ID: 7035
User: MOOGLE\cgiddin
Computer: MOOGLE

Description:

The IBM MQSeries Broker ConfigMgr service was successfully sent a start control.

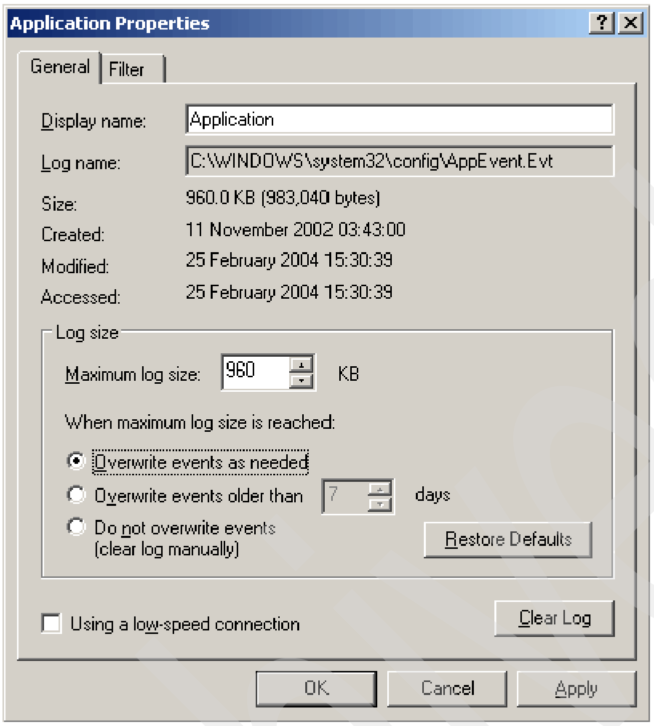
For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Data: [✓ Bytes] Words

*Figure 6-15 System Log message response to mqsistart configmgr command*

It is important to ensure that the properties for the Application log and System log have been set up correctly in the Windows Event Viewer; otherwise, messages could fail to be recorded, which may lead to difficulties in troubleshooting problems in the runtime. The Application Log in particular can fill up very quickly if there are problems in the system, and either become full (in which case no messages are written) or begin overwriting messages. Either way, the original causes of a problem can be difficult to determine. To check the properties of the Application Log:

1. In the Windows Event Viewer, right-click the **Application** log in the left-side pane and select **Properties** from the context menu.
2. Increase the **Maximum log size**. This determines the number of messages that can be retained by the log. The size of log that is required depends on the amount of activity that is likely to be recorded over time. In this example (Figure 6-16 on page 178), the log size is `960 KB`. It has recorded just over 3,000 events.


Application Properties

General | Filter

Display name: Application

Log name: C:\WINDOWS\system32\config\AppEvent.Evt

Size: 960.0 KB (983,040 bytes)

Created: 11 November 2002 03:43:00

Modified: 25 February 2004 15:30:39

Accessed: 25 February 2004 15:30:39

Log size

Maximum log size: 960 KB

When maximum log size is reached:

[✓ Overwrite events as needed]

Overwrite events older than 7 days

Do not overwrite events (clear log manually)

*Figure 6-16 Application log Properties*

3. Select an action under **When maximum log sized is reached**. Selecting the first option, Overwrite events as needed, is recommended, as this prevents the log becoming full and the loss of recent messages. However, the option that is selected must take into account local working practices and how long event messages may have be retained. Clearing the log manually means that no old messages are lost, but it does run the risk of the log becoming full and new messages not being recorded.

The log can be manually cleared at any time by clicking the Clear Log button. The Application log can also be saved by right-clicking the Application log and selecting **Save log file as** from the context menu.

### 6.1.3 Message Brokers Toolkit Event Log

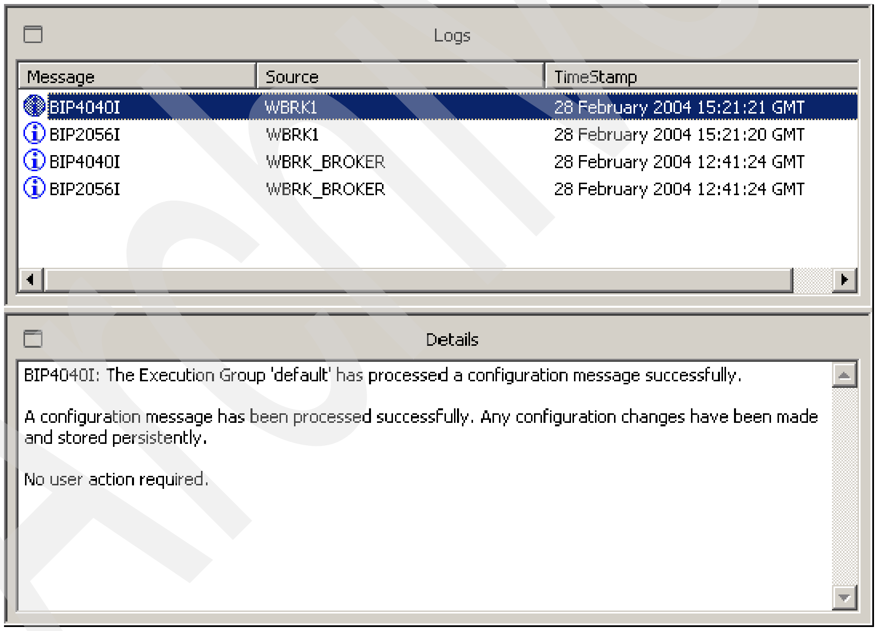
The Event Log editor in the Message Brokers Toolkit is the primary source of messages relating to deployment actions from the Message Brokers Toolkit to the Configuration Manager and the Broker Topology. The messages that are displayed in the Event Log are not recorded in the Windows Event Viewer, and the two logs must not be confused.

Whenever a deployment action is initiated through the Message Brokers Toolkit, the responses that are received are stored in the Configuration Repository database in the Configuration Manager. These messages are then displayed in the Event Log editor in the Message Brokers Toolkit.

To access the Event Log, double-click **Event Log** (marked with a flag icon) in the Domains view of the Broker Administration perspective in the Message Brokers Toolkit.

To view more information about the messages displayed in the Event Log, select the message from the top of the Event Log editor under Logs to display the details in the Details section. These are BIP messages with the same structure as those seen in other locations.

A deploy response message is seen for each broker that is involved in the deploy action in the Event Log, and the name of the broker is seen in the Source section under Logs. Figure 6-17 shows successful deploy messages for the brokers WBRK_BROKER and WBRK1.



| Message | Source | TimeStamp |
|---|---|---|
| BIP4040I | WBRK1 | 28 February 2004 15:21:21 GMT |
| BIP2056I | WBRK1 | 28 February 2004 15:21:20 GMT |
| BIP4040I | WBRK_BROKER | 28 February 2004 12:41:24 GMT |
| BIP2056I | WBRK_BROKER | 28 February 2004 12:41:24 GMT |

BIP4040I: The Execution Group 'default' has processed a configuration message successfully.

A configuration message has been processed successfully. Any configuration changes have been made and stored persistently.

No user action required.

*Figure 6-17 Messages displayed in the Event Log in the Message Brokers Toolkit*

When an error message is displayed in the Message Brokers Toolkit Event Log, it may be necessary to look for other, related error messages in the Windows Event Viewer or a remote systems local log to determine the cause of the failure.

The Event Log updates itself automatically as messages are stored in the Configuration Repository, but right-clicking in the Event Log and selecting Revert refreshes the information in the log. Using this context menu, you also can filter and clear the log. Using Clear removes the messages from the Configuration Repository database, so this must be used only if the messages do not have to be kept. When messages are present in the log, then the log can also be saved from the context menu as text files.

## 6.1.4 Eclipse Error Log

The Message Brokers Toolkit is built on the Eclipse platform, and errors that occur with the Message Brokers Toolkit itself are displayed in the Eclipse PDE Runtime Error Log. To display the PDE Runtime Error Log in any perspective:

1. From the **Window** menu, select **Show View**.
2. Select **Other** from the list of options.
3. In the Show View window, expand the **PDE Runtime** folder and select **Error Log** (Figure 6-18).
4. Click on **OK**.

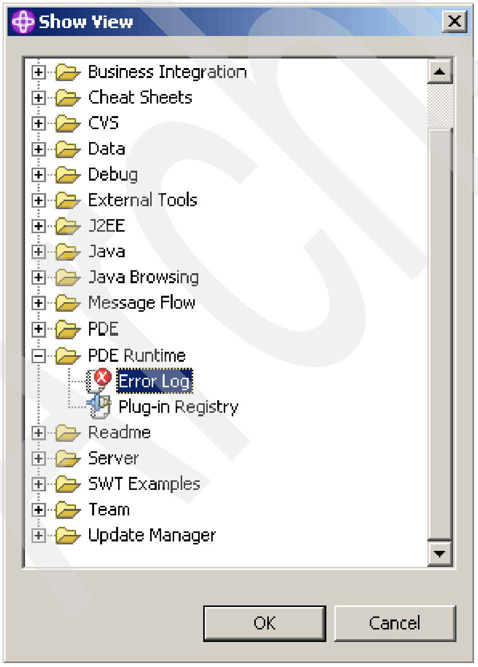
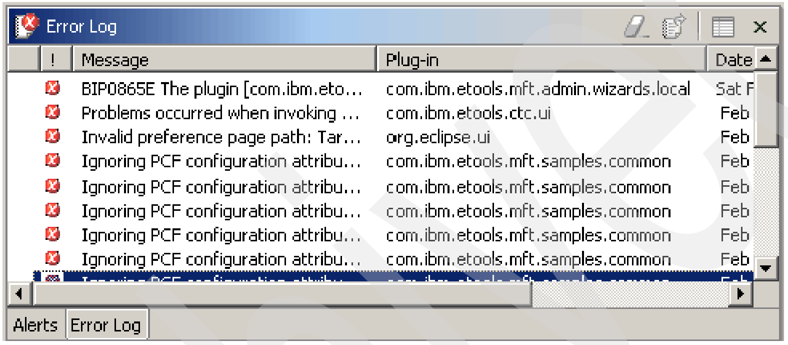


*Figure 6-18 Viewing the PDE Runtime Error Log*

The PDE Runtime Error Log tab is usually displayed at the bottom of a perspective, along with Tasks or Alerts. This log is a visual display of the error messages that can be seen in the .log file in the .metadata directory of the Workspace directory in the Eclipse directory structure. These error messages are generated by Eclipse and the Message Brokers Toolkit, and not all of the error messages displayed here are of concern to a user of the Message Brokers Toolkit, but some BIP messages can be seen here.

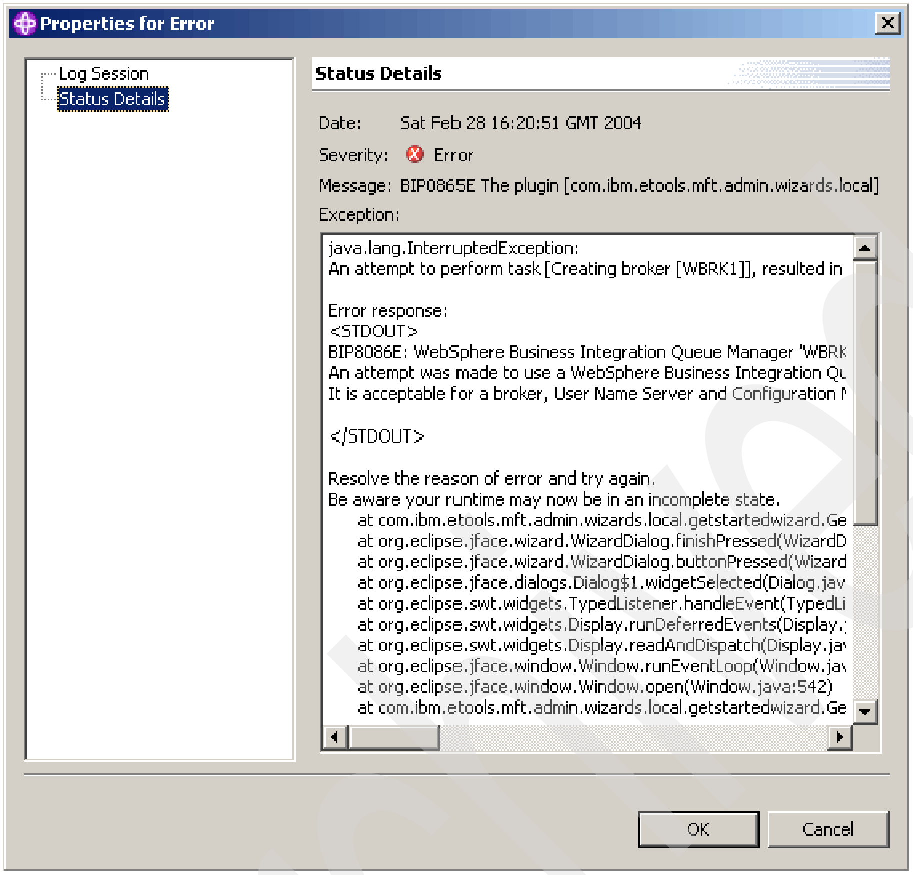
The PDE Runtime Error Log is shown in Figure 6-19. The BIP message displayed here is the result of an error that occurred during component creation by the Getting Started Wizard.


Error Log

| ! | Message | Plug-in | Date |
|---|---|---|---|
| ✖ | BIP0865E The plugin [com.ibm.eto... | com.ibm.etools.mft.admin.wizards.local | Sat F |
| ✖ | Problems occurred when invoking ... | com.ibm.etools.ctc.ui | Feb |
| ✖ | Invalid preference page path: Tar... | org.eclipse.ui | Feb |
| ✖ | Ignoring PCF configuration attribu... | com.ibm.etools.mft.samples.common | Feb |
| ✖ | Ignoring PCF configuration attribu... | com.ibm.etools.mft.samples.common | Feb |
| ✖ | Ignoring PCF configuration attribu... | com.ibm.etools.mft.samples.common | Feb |
| ✖ | Ignoring PCF configuration attribu... | com.ibm.etools.mft.samples.common | Feb |
| ✖ | Ignoring PCF configuration attribu... | com.ibm.etools.mft.samples.common | Feb |

*Figure 6-19 BIP error message in the PDE Runtime Error Log*

To display the details of an error message in the PDE Runtime Error Log, double-click the error message of interest. Some error messages have more detail about the error in the Status Details section of the error properties, as shown for BIP0865E in Figure 6-20 on page 182.



```
Date:      Sat Feb 28 16:20:51 GMT 2004
Severity:  Error
Message:   BIP0865E The plugin [com.ibm.etools.mft.admin.wizards.local]
Exception:
java.lang.InterruptedException:
An attempt to perform task [Creating broker [WBRK1]], resulted in

Error response:
<STDOUT>
BIP8086E: WebSphere Business Integration Queue Manager 'WBRK
An attempt was made to use a WebSphere Business Integration Qu
It is acceptable for a broker, User Name Server and Configuration M

</STDOUT>

Resolve the reason of error and try again.
Be aware your runtime may now be in an incomplete state.
    at com.ibm.etools.mft.admin.wizards.local.getstartedwizard.Ge
    at org.eclipse.jface.wizard.WizardDialog.finishPressed(WizardD
    at org.eclipse.jface.wizard.WizardDialog.buttonPressed(Wizard
    at org.eclipse.jface.dialogs.Dialog$1.widgetSelected(Dialog.jav
    at org.eclipse.swt.widgets.TypedListener.handleEvent(TypedLi
    at org.eclipse.swt.widgets.Display.runDeferredEvents(Display.j
    at org.eclipse.swt.widgets.Display.readAndDispatch(Display.jav
    at org.eclipse.jface.window.Window.runEventLoop(Window.jav
    at org.eclipse.jface.window.Window.open(Window.java:542)
    at com.ibm.etools.mft.admin.wizards.local.getstartedwizard.Ge
```

*Figure 6-20 Details of error message generated by Getting Started wizard*

In this example the error has been generated by attempting to create a broker using the Getting Started Wizard, which uses a queue manager that is already being used by another broker. The information that is contained in the STDOUT section was the information and error message produced by the WebSphere Business Integration Message Brokers runtime when the Getting Started Wizard tried to create the broker with the values supplied by the user.

This information is not recorded elsewhere, and it is very useful in tracking down the cause of the failure.

### 6.1.5 Other useful logs

A variety of other logs that record information are created when using WebSphere Business Integration Message Brokers. This section gives a brief description of some of the other logs that are generated.

#### Install Log

Installation of WebSphere Business Integration Message Brokers on Windows creates these logs:

- mqsi5.log
- mqsi5_enroll.log
- mqsi5_envvar.log
- mqsi5_getcap.log
- mqsi5_SecurityWiz.log
- mqsi5_setcap.log

The main log for the installation is the mqsi5.log. The other logs contain information relating to the setting of environment variables, licensing, and the Security Wizard.

#### MRM logs

A number of logs are generated in relation to message sets by WebSphere Business Integration Message Brokers. Logs are generated in Message Set Projects during the import or export of message definitions such as XML schema. A log is also created by the runtime if deployment of a message set with a TDS layer fails. This type of error is indicated by a BIP1836 message in the Message Brokers Toolkit Event Log.

## 6.2 Using the message flow debugger

In order to use the message flow debugger, the Agent Controller must be installed and running on the system where the flow is deployed. Breakpoints must be added to the flows in order to track the progress and status of any messages passing through the flow.

The message flow debugger is useful for determining what actually happens in a flow, and for tracking down problems or unexpected behavior. It is especially useful in complex flows, where the route of messages can be tracked and input and output messages can be verified through the flows.

In this section, the Error Handler sample is used to demonstrate the message flow debugger. We use this sample because it contains a variety of nodes, including ESQL and a subflow.

The message flow debugger has its own perspective in the Message Brokers Toolkit which can be accessed by choosing Flow Debug from the Open Perspective command on the Window menu.

## 6.2.1 Adding breakpoints to a message flow

A breakpoint is a point defined between nodes in a message flow at which point progress of a message is stopped by the message flow debugger so that the message can be examined and altered. If no breakpoints are created in a flow, then a message passes through the flow without being stopped, so it cannot be examined.

There are several ways to add breakpoints to connections between nodes:

1. Right-click a connection between two nodes and select **Add Breakpoint** from the context menu.
2. Right-click a node and select **Add Breakpoints Before Node** from the context menu (Figure 6-21) to add a breakpoint on all connections that enter the node.
3. Right-click a node and select **Add Breakpoints After Node** (also in Figure 6-21) to add a breakpoint on all connections that leave the node.

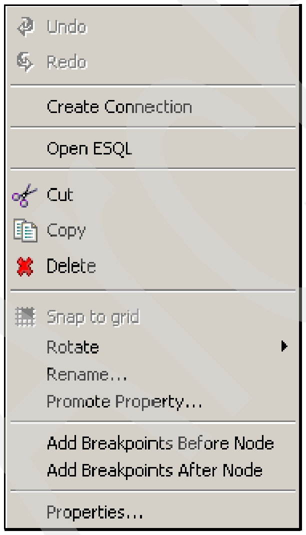


*Figure 6-21 Message flow context menu*

If it is known that the problem that is under investigation occurs between a particular set of nodes, then breakpoints can be added only in those locations. For tracking the entire progress of a test message, all of the connections must be given breakpoints. A subflow can also be given breakpoints (Figure 6-22). If no breakpoints are created in a subflow, then the message flow debugger does not stop progress at a subflow, and the message is only tracked going into and out of the subflow, but not through it.

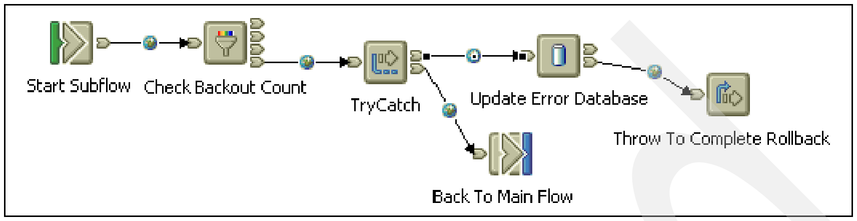


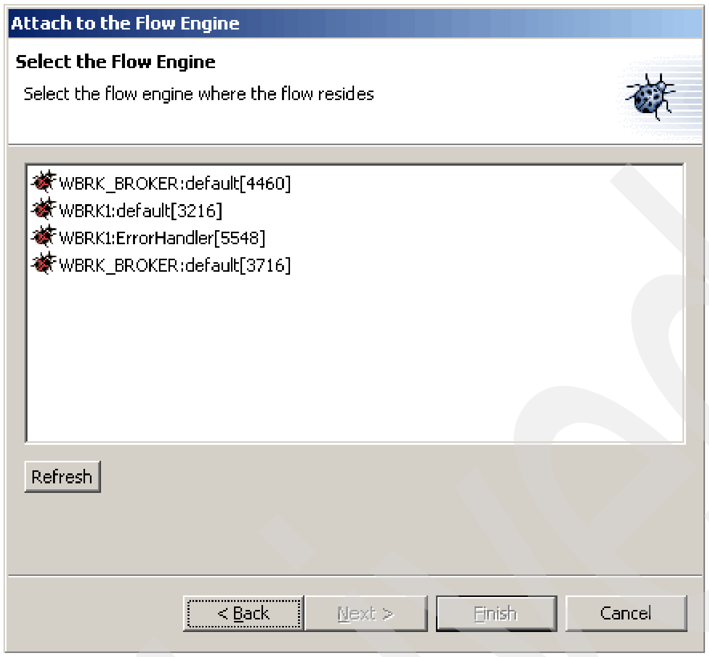
*Figure 6-22 Message flow showing breakpoints on the connections*

A deployed message flow can have breakpoints added to it without the need to redeploy. This is different from the message flow debugger in previous versions of the WebSphere Business Integration Message Brokers product, as it used to be necessary for a deploy to take place after breakpoints were added. This means that breakpoints can be added and removed without having to redeploy the flows.

## 6.2.2 Attaching the message flow debugger

The message flow debugger must be attached to a specific execution group in order to start debugging message flows. This means that any flows that are running in that execution group, including subflows, can be debugged at the same time. To attach the message flow debugger:

1. In the Flow Debug perspective, click the Attach to Flow Runtime button on the toolbar.
2. On the first page of the Attach to the Flow Engine wizard, select or enter the name or IP address of the machine where the execution group is running. Select **localhost** if it is the same system on which the Message Broker Toolkit is running. Click **Next**.
3. On the next page of the Attach to the Flow Engine wizard, select the execution group to which to attach the message flow debugger and click **Finish**. Figure 6-24 on page 187 shows an example of how the execution groups are displayed in the wizard: broker name, followed by execution group name, followed by execution group process ID in square brackets.


Attach to the Flow Engine

Select the Flow Engine

Select the flow engine where the flow resides

WBRK_BROKER:default[4460]
WBRK1:default[3216]
WBRK1:ErrorHandler[5548]
WBRK_BROKER:default[3716]

*Figure 6-23 Execution groups in the Attach to Flow Engine wizard*

The Message flows deployed to the selected execution group are then displayed in the Flow Debug view in the Message Brokers Toolkit.

**Note:** The Flow Debugger can be attached to multiple execution groups.

### 6.2.3 Tracking a message through a flow

The next stage in using the message flow debugger is to put a test message through a message flow. This task can be performed using any tool to put a message to an input node. In this example, the Enqueue message utility in the Message Brokers Toolkit is used to put a test message to the input queue of the Error Handler message flow. The Error Handler sample has been deployed, the runtime resources have been created, breakpoints have been added to the Main_flow and Error_Handler flow, and the message flow debugger has been attached to the execution group to which the sample has been deployed.

1. Create a new **Message Enqueue** file.
2. Specify the correct queue manager and queue name in the appropriate fields (default queue manager is WBRK_QM and default queue name for the Error Handler sample is STAFF_IN).

3. Browse for a test message to put to the queue (for the Error Handler sample, use the valid message staffmsg.txt).
4. Click **Write To** to put the test message to the input queue.

The flow debugger opens the message flow to which the message has been put and highlights the first connection with a breakpoint to indicate the progress of the message. Figure 6-25 shows this for the Main_Flow Error Handler sample.

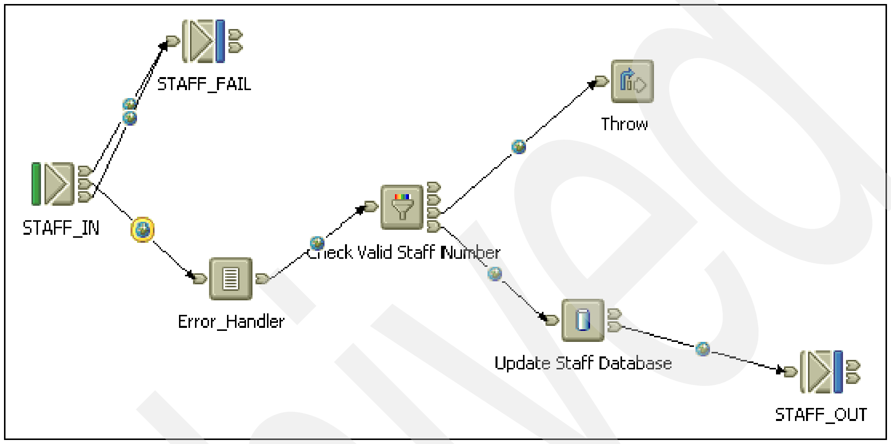


*Figure 6-24 Message at a breakpoint in the Message Flow Debugger*

The content of the message at this point can also be viewed by expanding the message in the Flow Debug Message view on the right side of the Flow Debug perspective (Figure 6-26 on page 188).

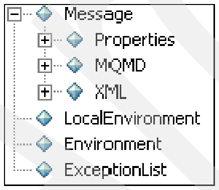


*Figure 6-25 Flow Debug message*

To allow the message to progress through the flow, click the Resume Flow Execution (the green arrow) button on the toolbar in the Flow Debug view. This can be clicked continually until the message exits the flow, and the progress and structure of the message can be monitored throughout.

When the message passes through the subflow, if the subflow also contains breakpoints then its progress also stops in the subflow until flow execution is permitted to continue.

## 6.2.4 Stepping through ESQL

When a message that is passing through a message flow in the message flow debugger is stopped at a breakpoint on a connection before a node containing ESQL, an arrow with a square appears over the top of the node (Figure 6-27 on page 192, which shows a breakpoint before a Filter node). This arrow means that the message flow debugger can step into the ESQL code at this point.

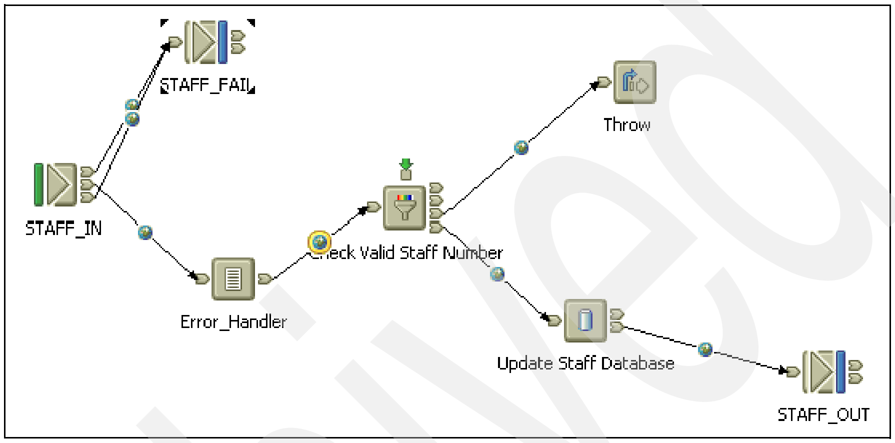


*Figure 6-26 Arrow icon over Filter node*

To step into the ESQL, click the Step Into Source Code button on the toolbar in the Flow Debug view. (This button has the same arrow with a square as is seen over the node containing ESQL.)

This shifts to the Debug perspective, which is used in Eclipse for debugging code. The controls and views are similar to the Flow Debug perspective in the Message Brokers Toolkit. The arrows on the toolbar of the debug view can be used to step line by line through the ESQL. The Variables view shows the debug message; where the ESQL is constructing an output message, such as in a compute node, the message can be seen to build up line by line by the ESQL. Try using this feature in the message flow debugger in a more complex message flow example such as the Airline sample.

After the ESQL code is completed, the message and view return to the standard message flow debugger to continue through the message flow.

## 6.2.5 Flow of errors in a message flow

In an ordinary message flow, the progress of the message is from the Input node through the message flow in a downstream direction. However, when an error occurs, the flow changes and messages start to move upstream as they begin to roll back. If error-handling messages are not added to a message flow, then when errors occur the message eventually rolls back to the input node and even back onto the input queue.

The Error Handler sample provides useful examples of how to handle errors in a message flow. It is also helpful to look at message flows by using the debugger, as this shows how messages are handled and rolled back when an error occurs. To demonstrate this, follow these instructions to input an invalid message to the Error Handler sample and use the debugger to see how the exceptions are handled within the flow:

1. Create an **Enqueue Message** file and set the correct queue manager name and the queue name (by default this is STAFF_IN for the Error Handler sample.
2. Browse for the invalidstaffmsg.txt file and click **Write To** to put the message in the queue.
3. In the Flow Debug perspective, pass the message through the flow until it reaches the Check Valid Staff Number node. When the message passes through this Filter node, it exits through the False terminal to a Throw node.
4. Pass the message to the Throw node. The message is rolled back through the message throw to a TryCatch node in the subflow. The cross in a red box over the top of the TryCatch node indicates that the message has rolled back to this point.
5. Check the message structure in the Flow Debug Message view. Several errors are seen in the ExceptionList. The message at the bottom of the tree is the user exception that was generated by the Throw node to indicate the problem with the message.
6. Resume flow execution. The message passes out of the Catch terminal of the TryCatch node, adds an entry to the error database through the Update Error Database node, and when flow execution is resumed again is sent to another Throw node.
7. Resume flow execution from the Throw node. The message is rolled back to the TryCatch node.
8. Resume flow execution again, and the message is rolled upstream to the input node. As the input node failure terminal is connected, the message is put to a failure queue. If the failure terminal was not connected, the message is backed out to the input queue.

When using the message flow debugger, all messages should be passed out of the flow before disconnecting the message flow debugger. It is also not possible to deploy to an execution group when the debugger is connected to it.

To disconnect the message flow debugger: In the Flow Debug view of the Flow Debug perspective, right-click the message flow debugger connections and select **Detach** from the context menu.

## 6.3 Using trace

This section gives an overview of using trace with WebSphere Business Integration Message Brokers. Trace is used when a problem occurs that cannot be solved using any of the methods that have been described so far in this chapter. Trace is used most often on execution groups, where it can be used to work out exactly what is happening when a message passes through a message flow. Message flows can even be designed to output user-specified trace, such as the content of the message tree using a Trace node.

Trace can also be used for helping to debug problems with WebSphere Business Integration Message Brokers commands or components, although this is usually only used for the purpose of collecting trace for service if a potential product defect is occurring. In addition, you also can start and collect trace from components of the Message Brokers Toolkit.

The types of trace that are described here are set up and collected only when necessary, as tracing results in a decrease in performance due to the extra processing that is required in a system.

For tracing on execution groups, components, and commands, there are different types and levels of trace and amounts of detail: user trace, service trace, and normal and debug levels of trace. Of these combinations, user trace at normal level has the least information, and service trace at debug level has the greatest amount of information. However, most of the information that is contained in the service trace is not of value to most users.

### 6.3.1 Tracing execution groups

Tracing of execution groups to obtain detailed information about what happens to messages that pass through a flow is straightforward and very useful. The Error Handler sample is used here, as above, to demonstrate how to perform trace on an execution group, and to show how to read the output from a trace file.

In order to follow the instructions in this example, the following tasks must have been performed (refer to earlier chapters of the book). The simplest way to do this is to use the cheat sheet.

- Import the Error Handler sample.
- Create the runtime resources (MQ queues and DB2 databases) for the Error Handler sample.
- Deploy the Error Handler sample.
- Create an **Enqueue Message** file to pass a test message through the flow.

Before you run any trace, all existing trace files should be deleted from the system. Although trace is not active by default, some files are created during normal operation of the WebSphere Business Integration Message Brokers runtime. Most trace files are stored in the log directory in the install location of the product. (You can change this location during the creation of components by specifying the -w parameter on the command line.) To delete the existing trace files on the system:

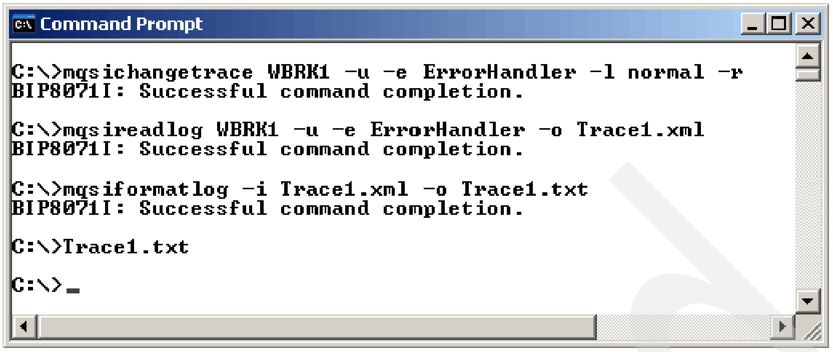
1. Stop all running WebSphere Business Integration Message Brokers components using `mqsistop` commands. Type `mqsilist` on the command line to determine which components are on the system if required.
2. Navigate to the \log directory in the install path of the product (for example: C:\Program Files\IBM\WebSphere Business Integration Message Broker).
3. Select all files and delete them.
4. Restart the components using `mqsistart`.
5. Ensure that the components started sucessfully.

### User trace level normal

This section traces messages through the Error Handler message flow using user trace at normal level.

This example uses these parameters: `WBRK1` for the broker and `ErrorHandler` for the execution group with the Error Handler sample deployed to it.

Figure 6-28 on page 195 shows the command line input and output of the exercise.



```
C:\>mqsichangetrace WBRK1 -u -e ErrorHandler -l normal -r
BIP8071I: Successful command completion.

C:\>mqsireadlog WBRK1 -u -e ErrorHandler -o Trace1.xml
BIP8071I: Successful command completion.

C:\>mqsiformatlog -i Trace1.xml -o Trace1.txt
BIP8071I: Successful command completion.

C:\>Trace1.txt

C:\>_
```

*Figure 6-27 Commands for user trace at normal level*

1. On the command line, enter the following command, substituting the broker name and execution group to the appropriate names:

   ```
   mqsichangetrace WBRK1 -u -e ErrorHandler -l normal -r
   ```

   **`Mqsichangetrace`** is the command that changes the trace settings on the broker name that follows it (in this case, `WBRK1`). The parameter `-u` indicates that the type of trace required is user trace, `-e` and the name following it are for the execution group to trace (in this case the execution group is called `ErrorHandler`), `-l` is the level of the trace, in this case normal, and `-r` is to reset the trace log. The `-r` parameter is not necessary here, as the log is clean, but it is useful to reset the log; otherwise, the trace file can become very large and difficult to read.

   The command results in a BIP8071 success message. If a failure occurs, it may be that one of the parameters was typed incorrectly. Use the details of the error message, if any, to work out the cause of any problem.

   The result of the command that has been run is to change the trace on the broker and execution group to which the Error Handler sample is deployed to user trace at normal level. This means that trace is now collected for certain actions that occur within this execution group. The next step is to put a message through the Error Handler message flow to generate some of these actions that are recorded in the trace.

2. Using the Enqueue Message file that you created, write the staffmsg.txt message from the Error Handler sample onto the STAFF_IN queue to put the message through the message flow.
3. Verify that a message appeared on the STAFF_OUT queue as expected.
4. Type the following command onto the command line to read the contents of the trace log into an XML file, substituting the names of the broker and execution group:

   ```
   mqsireadlog WBRK1 -u -e ErrorHandler -o Trace1
   ```

The **mqsireadlog** command reads the contents of the trace log into an XML file. The other parameters that are given with the command determine which trace log the information is being read from, as many components, each with its own trace log, may be being traced. In this example, the name that follows the **mqsireadlog** command is the broker name, the `-u` takes the user trace information from the trace log, `-e` and the name following it are the execution group being traced, and the `-o` parameter is the name of the output file to put the trace information to. This does not require a file extension, but it is useful to give it an .xml file extension so that it is recognizable as an XML file rather than a fully formatted trace file.

At this stage it is possible to locate and view the trace file, but it is in XML format and is difficult to read. Therefore, a further step must be carried out to get the file into a formatted state for interpretation. This can be performed using the **mqsiformatlog** command.

5. Type the following command onto the command line:

   ```
   mqsiformatlog -i Trace1.xml -o Trace1.txt
   ```

   The **mqsiformatlog** command converts an XML file that was generated by the **mqsireadlog** command into a formatted file that can be read by a text editor. It has two parameters: -i is the name of the XML input file, and -o is the name to give to the output file. In Windows, it is useful to give the file a .txt file extension so that the file can be opened from the command line into the Notepad program simply by typing the name of the file on the command line.

6. Open the trace file that was generated by the **mqsiformatlog** command (in this example, called `Trace1.txt`). If Notepad is used as the text editor, ensure that Word Wrap is *not* on, as this makes reading the trace easier.

The output in the trace file consists of several lines of usertrace, each with a timestamp at the beginning, followed by a four-character code, the type of trace (in this example all messages are `UserTrace`), a BIP code, and a description of the event with details such as node names, queue names, and parser types. This gives details of all events that occur as the message passes through the message flow.

Example 6-1 shows the content of the trace file that was generated in the previous exercise with all of the extra information (except for the BIP information). The details of the configuration change were on the execution group as it started up, but this has also been removed for this example as it does not contain relevant information for the message that is traveling through the flow.

*Example 6-1 Excerpt from Trace1.txt file*

```
BIP2632I: Message received and propagated to 'out' terminal of MQ input node
'Main_Flow.STAFF_IN'.
```

```
BIP6060I: Parser type 'Properties' created on behalf of node
'Main_Flow.STAFF_IN' to handle portion of incoming message of length 0 bytes
beginning at offset '0'.

BIP6061I: Parser type 'MQMD' created on behalf of node 'Main_Flow.STAFF_IN' to
handle portion of incoming message of length '364' bytes beginning at offset
'0'. Parser type selected based on value 'MQHMD' from previous parser.

BIP6061I: Parser type 'XML' created on behalf of node 'Main_Flow.STAFF_IN' to
handle portion of incoming message of length '133' bytes beginning at offset
'364'. Parser type selected based on value 'XML' from previous parser.

BIP4004I: Message propagated to 'true' terminal of filter node 'Main_Flow.Check
Backout Count'.

BIP4080I: Message propagated to try terminal from try-catch node
'Main_Flow.Error_Handler.TryCatch'. The try-catch node
'Main_Flow.Error_Handler.TryCatch' has received a message and is propagating it
to any nodes connected to its try terminal. No user action required.

BIP4004I: Message propagated to 'true' terminal of filter node 'Main_Flow.Check
Valid Staff Number'.

BIP4184I: Message propagated to 'out' terminal of database node
'Main_Flow.Update Staff Database'.

BIP2638I: The MQ output node 'Main_Flow.STAFF_OUT' attempted to write a message
to queue 'STAFF_OUT' connected to queue manager ''. The MQCC was '0' and the
MQRC was '0'.

BIP2622I: Message successfully output by output node 'Main_Flow.STAFF_OUT' to
queue 'STAFF_OUT' on queue manager ''.
```

Perform this exercise again as before, but this time use the invalidstaffmsg.txt file to generate an error when passing the message through the message flow, using the following instructions and substituting the broker name and execution group where appropriate (Figure 6-28 on page 195 shows the commands that were used):

1. On the command line, enter:

   ```
   mqsichangetrace WBRK1 -u -e ErrorHandler -l normal -r
   ```

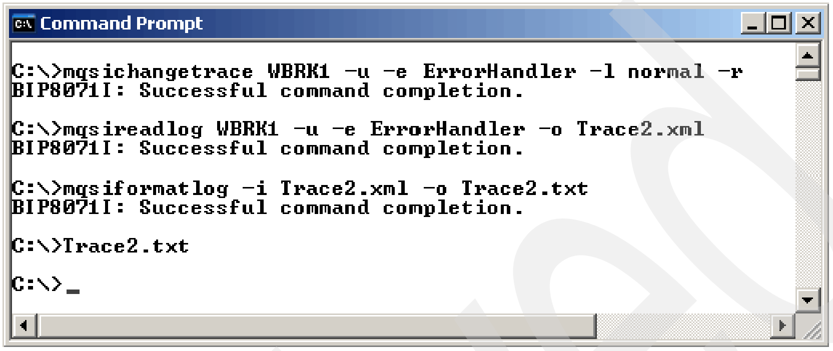
2. Using the Enqueue Message file that you created, write the `invalidstaffmsg.txt` message from the Error Handler sample onto the `STAFF_IN` queue.
3. Verify that a message appeared on the STAFF_FAIL queue as expected.
4. Type the following command onto the command line to read the contents of the trace log into an XML file:

   ```
   mqsireadlog WBRK1 -u -e ErrorHandler -o Trace2.xml
   ```

5. Type the following command onto the command line:

```
mqsiformatlog -i Trace2.xml -o Trace2.txt
```

6. Open the trace file generated by the **mqsiformatlog** command (in this example called `Trace2.txt`).



```
C:\>mqsichangetrace WBRK1 -u -e ErrorHandler -l normal -r
BIP8071I: Successful command completion.

C:\>mqsireadlog WBRK1 -u -e ErrorHandler -o Trace2.xml
BIP8071I: Successful command completion.

C:\>mqsiformatlog -i Trace2.xml -o Trace2.txt
BIP8071I: Successful command completion.

C:\>Trace2.txt

C:\>_
```

*Figure 6-28 Commands for user trace at normal level second trace file*

The trace that was produced by putting the invalidstaffmsg.txt file to the Error Handler message flow shows a different chain of events than the previous trace exercise. Instead of UserTrace, the word `Error` appears in front of the BIP2232 error message. This message is in fact displayed in the Application Log in the Windows Event Viewer, and it is the only BIP message recorded in either of these trace files that is displayed in the Event Viewer. There is also a UserException in front of the BIP3002 message where a user exception has been generated as part of the error handling in the flow itself. The changes are shown in the trace excerpt in Example 6-2, again just showing the BIP codes and descriptions. The exceptions produced by the invalid message are clearly seen.

*Example 6-2 Excerpt from Trace2.txt*

```
BIP4005I: Message propagated to 'false' terminal of filter node
'Main_Flow.Check Valid Staff Number'.

BIP4101I: Exception thrown by throw node 'Main_Flow.Throw'. The throw node
'Main_Flow.Throw' has received a message and will throw an exception as this is
its normal behavior. No user action required.

BIP4081I: Message propagated to catch terminal from try-catch node
'Main_Flow.Error_Handler.TryCatch'. The try-catch node
'Main_Flow.Error_Handler.TryCatch' has caught an exception which occurred in a
node connected to its try terminal. The message has been augmented with an
exception list and is propagating it to any nodes connected to its catch
terminal for further processing. See the following messages for details of the
exception list. No user action required.
```

```
BIP3001I: Exception thrown by throw node 'Main_Flow.Throw'; text is 'Invalid
staff number'. The throw node 'Main_Flow.Throw' has received a message and thus
has thrown an exception as this is its normal behavior. The message text
associated with this exception is 'Invalid staff number'. Since this is
application generated (by message flow behavior), the user action is determined
by the message flow and the type of exception generated.

BIP4184I: Message propagated to 'out' terminal of database node
'Main_Flow.Update Error Database'.

BIP4101I: Exception thrown by throw node 'Main_Flow.Error_Handler.Throw To
Complete Rollback'. The throw node 'Main_Flow.Error_Handler.Throw To Complete
Rollback' has received a message and will throw an exception as this is its
normal behavior. No user action required.

BIP2232E: Error detected whilst handling a previous error in node
'Main_Flow.Error_Handler.Throw To Complete Rollback'. The message broker has
detected an error in node 'Main_Flow.Error_Handler.Throw To Complete Rollback'
whilst handling a previous error. See the following messages for details of the
exception list associated with the original error. Thereafter messages will be
associated with the new error.

BIP3001I: Exception thrown by throw node 'Main_Flow.Throw'; text is 'Invalid
staff number'. The throw node 'Main_Flow.Throw' has received a message and thus
has thrown an exception as this is its normal behavior. The message text
associated with this exception is 'Invalid staff number'. Since this is
application generated (by message flow behavior), the user action is determined
by the message flow and the type of exception generated.

BIP2231E: Error detected whilst processing a message 'Main_Flow.STAFF_IN'. The
message broker detected an error whilst processing a message in node
'Main_Flow.STAFF_IN'. The message has been augmented with an exception list and
has been propagated to the node's failure terminal for further processing. See
the following messages for details of the error.

BIP3002I: Exception thrown by throw node 'Main_Flow.Error_Handler.Throw To
Complete Rollback'; text is 'From Error_Handler message flow. See ERRORDB for
details.'. The throw node 'Main_Flow.Error_Handler.Throw To Complete Rollback'
has received a message and thus has thrown an exception as this is its normal
behavior. The message text associated with this exception is 'From
Error_Handler message flow. See ERRORDB for details.'. Since this is
application generated (by message flow behavior), the user action is determined
by the message flow and the type of exception generated.

BIP2638I: The MQ output node 'Main_Flow.STAFF_FAIL' attempted to write a
message to queue 'STAFF_FAIL' connected to queue manager ''. The MQCC was '0'
and the MQRC was '0'.

BIP2622I: Message successfully output by output node 'Main_Flow.STAFF_FAIL' to
queue 'STAFF_FAIL' on queue manager ''.
```

You can see that this level of trace is sufficient for the majority of purposes, but in order to get even more information, including at the level of actions on a message with ESQL, then a higher level of trace is required.

### User trace level debug

To demonstrate user trace at debug level, the previous exercise is run as before, this time using the invalidstaffmsg.txt file. Follow these instructions, substituting the broker name and execution group where appropriate (Figure 6-29 shows the commands used in this exercise):

1. On the command line, enter:

   ```
   mqsichangetrace WBRK1 -u -e ErrorHandler -l debug -r
   ```

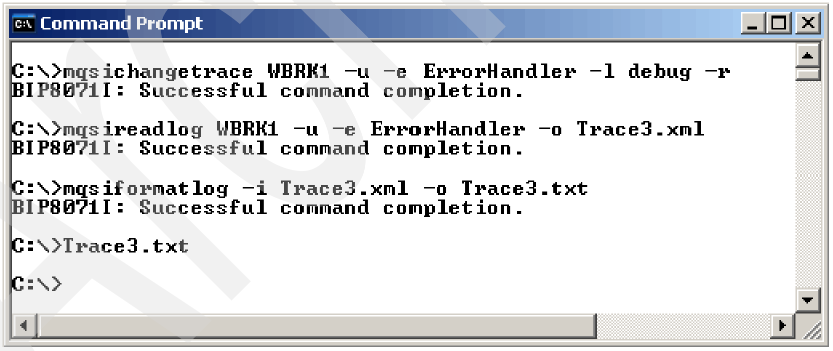
2. Using the Enqueue Message file that you created, write the invalidstaffmsg.txt message from the Error Handler sample onto the STAFF_IN queue.
3. Verify that a message appeared on the STAFF_FAIL queue as expected.
4. Type this command onto the command line to read the contents of the trace log into an XML file:

   ```
   mqsireadlog WBRK1 -u -e ErrorHandler -o Trace3.xml
   ```

5. Type this command onto the command line:

   ```
   mqsiformatlog -i Trace3.xml -o Trace3.txt
   ```

6. Open the trace file that was generated by the mqsiformatlog command (in this example called Trace3.txt).



```
C:\>mqsichangetrace WBRK1 -u -e ErrorHandler -l debug -r
BIP8071I: Successful command completion.

C:\>mqsireadlog WBRK1 -u -e ErrorHandler -o Trace3.xml
BIP8071I: Successful command completion.

C:\>mqsiformatlog -i Trace3.xml -o Trace3.txt
BIP8071I: Successful command completion.

C:\>Trace3.txt

C:\>
```

*Figure 6-29 Commands for user trace at debug level*

Upon viewing a debug trace file in comparison to a normal trace file, it is obvious that a lot more information is recorded in the debug trace file, making it harder to read but much clearer to work out exactly what is happening to a message in a message flow. A line is included for the evaluation of each line of ESQL that is run in the message flow. Example 6-3 on page 198 shows some examples from the trace file that was produced in this exercise to demonstrate how the error that occurs in this message flow can be seen in the trace file.

*Example 6-3 Excerpt from Trace3.txt*

```
BIP2537I: Node 'Main_Flow.Check Valid Staff Number': Executing statement 'IF
Body.Staff.StaffNumber <= '10' THEN... ELSE... END IF;' at
(.Main_Flow_Filter.Main, 3.2).

BIP2538I: Node 'Main_Flow.Check Valid Staff Number': Evaluating expression
'Body.Staff.StaffNumber <= '10'' at (.Main_Flow_Filter.Main, 3.27).

BIP2538I: Node 'Main_Flow.Check Valid Staff Number': Evaluating expression
'Body.Staff.StaffNumber' at (.Main_Flow_Filter.Main, 3.5).

BIP2539I: Node 'Main_Flow.Check Valid Staff Number': Finished evaluating
expression 'Body.Staff.StaffNumber <= '10'' at (.Main_Flow_Filter.Main, 3.27).
This resolved to ''99' <= '10''. The result was 'FALSE'.

BIP2537I: Node 'Main_Flow.Check Valid Staff Number': Executing statement
'RETURN FALSE;' at (.Main_Flow_Filter.Main, 6.3).

BIP4005I: Message propagated to 'false' terminal of filter node
'Main_Flow.Check Valid Staff Number'.

BIP4101I: Exception thrown by throw node 'Main_Flow.Throw'. The throw node
'Main_Flow.Throw' has received a message and will throw an exception as this is
its normal behavior. No user action required.

BIP4081I: Message propagated to catch terminal from try-catch node
'Main_Flow.Error_Handler.TryCatch'. The try-catch node
'Main_Flow.Error_Handler.TryCatch' has caught an exception which occurred in a
node connected to its try terminal. The message has been augmented with an
exception list and is propagating it to any nodes connected to its catch
terminal for further processing. See the following messages for details of the
exception list. No user action required.

BIP3001I: Exception thrown by throw node 'Main_Flow.Throw'; text is 'Invalid
staff number'. The throw node 'Main_Flow.Throw' has received a message and thus
has thrown an exception as this is its normal behavior. The message text
associated with this exception is 'Invalid staff number'. Since this is
application generated (by message flow behavior), the user action is determined
by the message flow and the type of exception generated.
```

## Service trace

Service trace can be collected using the same commands as user trace; the only difference is that the `-u` parameter in the **`mqsichangetrace`** and **mqsireadlog** commands is replaced by a `-t` parameter. An example is:

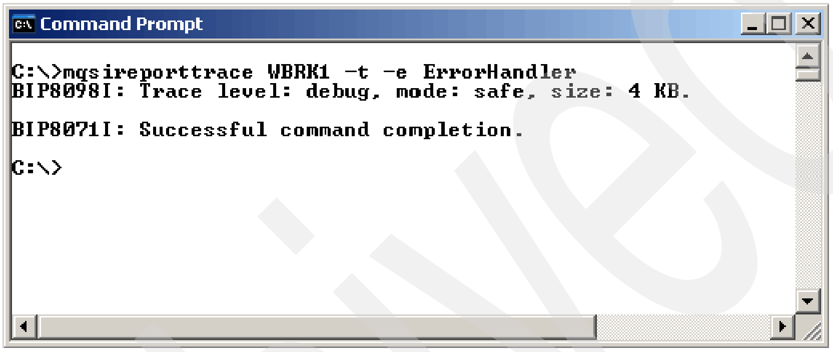
```
   mqsichangetrace WBRK1 -t -e ErrorHandler -l debug -r
```

## Display trace settings

The trace settings on an execution group or other WebSphere Business Integration Message Brokers can be displayed using the **mqsireporttrace** command. To display the service trace settings for the execution group used in the previous examples, enter the following on the command line, substituting the broker name and execution group name as necessary:

```
mqsireporttrace WBRK1 -t -e ErrorHandler
```

This command and the output can be seen in Figure 6-29 on page 197.



```
C:\>mqsireporttrace WBRK1 -t -e ErrorHandler
BIP8098I: Trace level: debug, mode: safe, size: 4 KB.

BIP8071I: Successful command completion.

C:\>
```

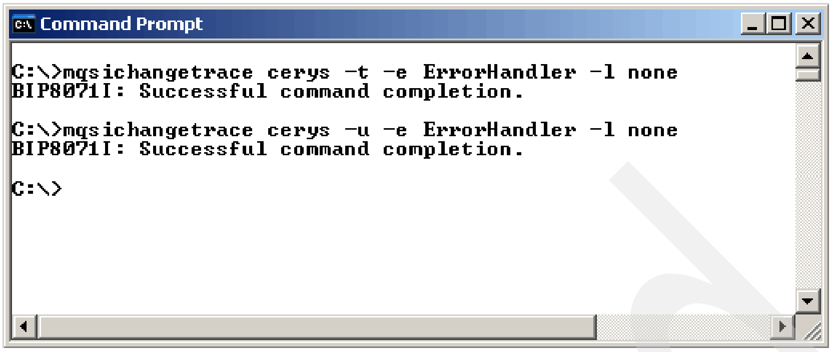
*Figure 6-30 Displaying trace levels using mqsireporttrace command*

To display the user trace settings for the execution group that was used in the previous examples, enter the following on the command line, substituting the broker name and execution group name as necessary:

```
mqsireporttrace WBRK1 -u -e ErrorHandler
```

## Reset trace settings

After trace has been collected, reset the trace settings to `none` to stop trace from being written. Figure 6-31 on page 200 shows the commands to set the user trace and the service trace to a level of none.



```
C:\>mqsichangetrace cerys -t -e ErrorHandler -l none
BIP8071I: Successful command completion.

C:\>mqsichangetrace cerys -u -e ErrorHandler -l none
BIP8071I: Successful command completion.

C:\>
```

*Figure 6-31 Resetting trace to none using mqsichangetrace command*

### 6.3.2 Tracing components

On occasion it may be necessary to collect trace for WebSphere Business Integration Message Brokers components, such as the Configuration Manager, the User Name Server, or brokers. These use the **mqsichangetrace** and **mqsireadlog** commands similar to trace on an execution group. Only service trace can be used for commands, so the `-t` parameter must be used with the commands, not the -u parameter.

To collect trace for the Configuration Manager:

1. Enter this command on the command line:

   ```
   mqsichangetrace configmgr -t -b -l normal
   ```

2. When trace is ready to be read, enter this command on a command line:

   ```
   mqsireadlog configmgr -t -b agent -f -o ConfigmgrTrace.xml
   ```

3. To format the log, enter the **`mqsiformatlog`** command as before, for example:

   ```
   mqsiformatlog -i ConfigmgrTrace.xml -o ConfigmgrTrace.txt
   ```

4. View the trace file that is generated.

The other runtime components of WebSphere Business Integration Message Brokers can be traced in the same way. This tracing is generally used in the event of collecting information for service.

### 6.3.3 Tracing commands

Only service trace can be used to trace WebSphere Business Integration Message Brokers commands. Trace for commands does not use the **mqsichangetrace** command; instead, only the **mqsireadlog** and **mqsiformatlog** commands are used as with the trace on components. However, an extra step must be performed to start trace for the commands by setting the MQSI_UTILITY_TRACE environment variable. The following instructions give an example of collecting trace for two commands, **mqsilist** and the **mqsistop** command on a broker.

To perform debug level service trace on the `mqsilist` command:

1. Open a command line and enter:

   ```
   set MQSI_UTILITY_TRACE=DEBUG
   ```

2. Enter:

   ```
   mqsilist
   ```

3. Enter:

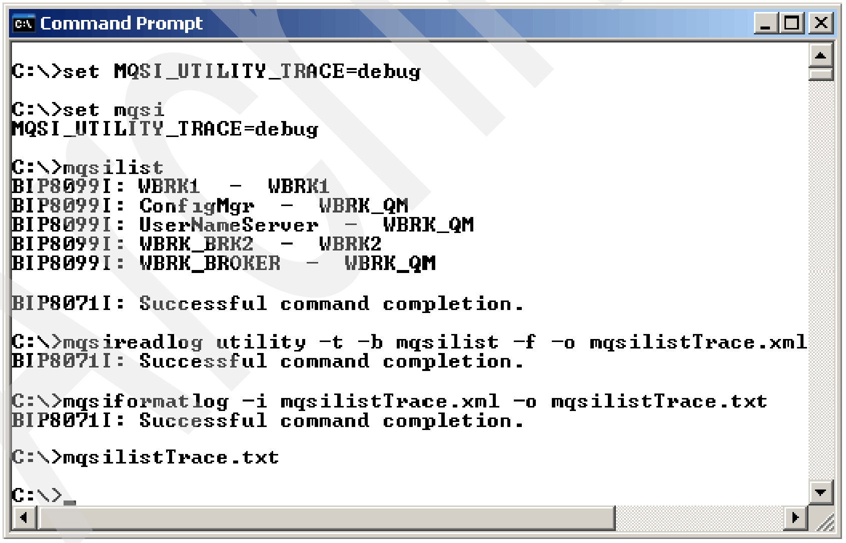
   ```
   mqsireadlog utility -t -b mqsilist -f -o mqsilistTrace.xml
   ```

4. Enter a **mqsiformatlog** command to format the trace for output to a text file; for example:

   ```
   mqsiformatlog -i mqsilistTrace.xml -o mqsilistTrace.txt
   ```

5. View the trace file.

Figure 6-32 shows the commands for tracing the `mqsilist` command and the output in the command-line session.



```
C:\>set MQSI_UTILITY_TRACE=debug

C:\>set mqsi
MQSI_UTILITY_TRACE=debug

C:\>mqsilist
BIP8099I: WBRK1   -  WBRK1
BIP8099I: ConfigMgr  -  WBRK_QM
BIP8099I: UserNameServer  -  WBRK_QM
BIP8099I: WBRK_BRK2  -  WBRK2
BIP8099I: WBRK_BROKER  -  WBRK_QM

BIP8071I: Successful command completion.

C:\>mqsireadlog utility -t -b mqsilist -f -o mqsilistTrace.xml
BIP8071I: Successful command completion.

C:\>mqsiformatlog -i mqsilistTrace.xml -o mqsilistTrace.txt
BIP8071I: Successful command completion.

C:\>mqsilistTrace.txt

C:\>_
```

*Figure 6-32 Tracing the mqsilist command*

To perform normal level service trace on the **mqsistop** command for the broker WBRK_BROKER:

1. On a command line, enter:

   ```
   set MQSI_UTILITY_TRACE=DEBUG
   ```

2. Enter:

   ```
   mqsistop WBRK_BROKER
   ```

3. Enter:

   ```
   mqsireadlog WBRK_BROKER -t -b mqsistop -f -o mqsistopTrace.xml
   ```

4. Enter a **mqsiformatlog** command to format the trace for output to a text file; for example:

   ```
   mqsiformatlog -i mqsistopTrace.xml -o mqsistopTrace.txt
   ```

5. View the trace file.

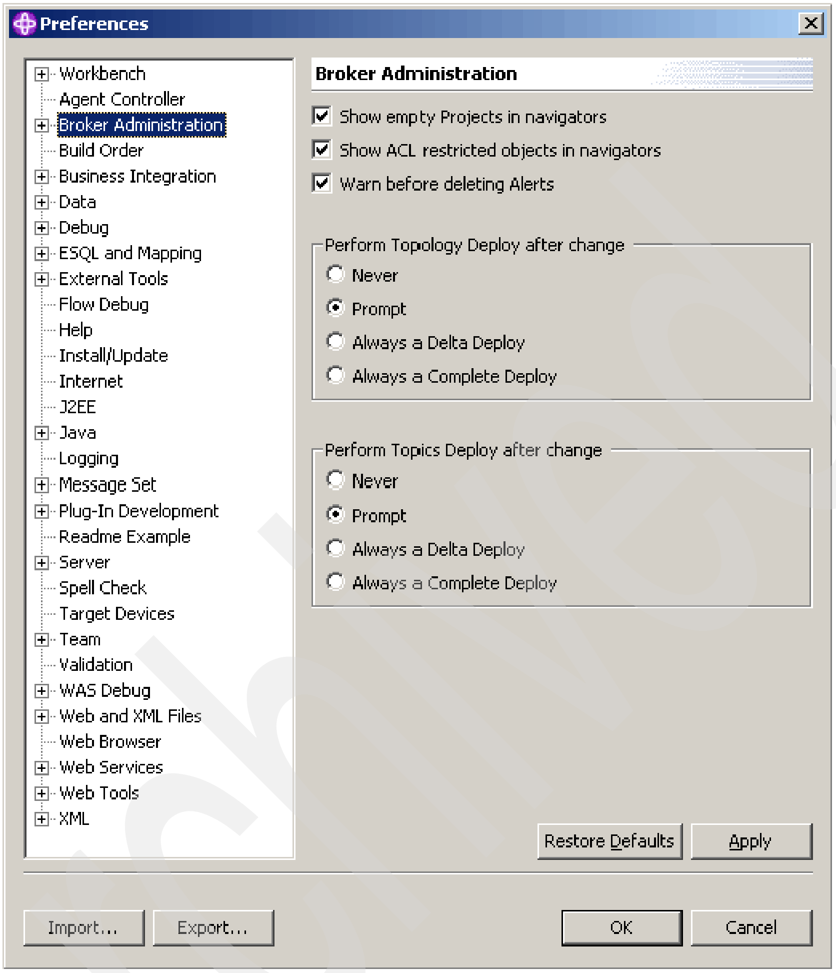
The advantage of setting the MQSI_UTILITY_TRACE environment variable on the command line rather than in the system is that at soon as the command line window is closed, the tracing is stopped. To manually stop the tracing of commands, type this on the command line:

```
set MQSI_UTILITY_TRACE=none
```

## 6.3.4 Tracing the Message Brokers Toolkit

Various components of the Message Brokers Toolkit in WebSphere Business Integration Message Brokers can have trace activated on them. This is not described in detail here, as trace on these components is only likely to be required if Service requests it. However, trace in the Message Brokers Toolkit and other advanced settings can be accessed through the Preferences window (in the Message Brokers Toolkit, select **Windows** → **Preferences**).

Figure 6-33 on page 203 shows the Preferences window with some of the settings that can be altered in the Broker Administration. It is in this section that most of the trace for the Message Brokers Toolkit can be configured as required.



**Broker Administration**

- [✓] Show empty Projects in navigators
- [✓] Show ACL restricted objects in navigators
- [✓] Warn before deleting Alerts

Perform Topology Deploy after change: Never [✓ Prompt] Always a Delta Deploy Always a Complete Deploy

Perform Topics Deploy after change: Never [✓ Prompt] Always a Delta Deploy Always a Complete Deploy

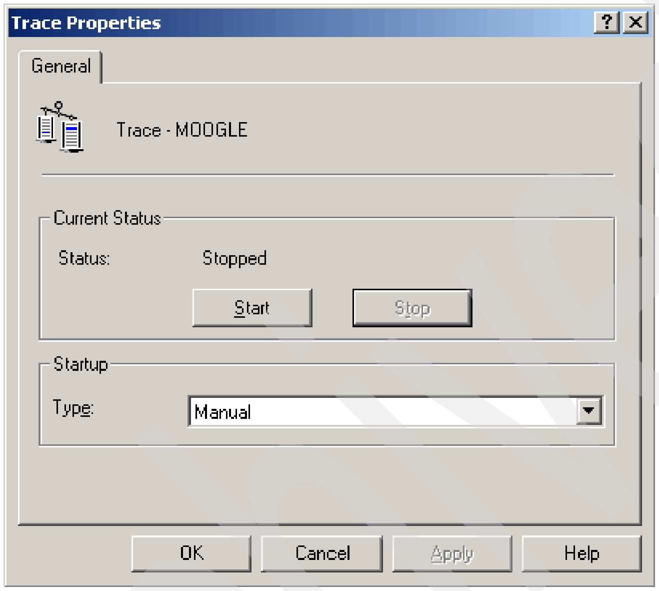
*Figure 6-33 Preferences in the Message Brokers Toolkit*

## 6.3.5 WebSphere MQ trace

Trace can be switched on within WebSphere MQ; however, this must be used with caution because it records trace for every executable running in WebSphere MQ and can lead to the production of very large log files and affect performance. Error logs are produced for WebSphere MQ without trace being set, and you should check these first if problems are occurring in WebSphere MQ before switching on trace. These error logs can be located in the Errors directory under the install path for WebSphere MQ (for example, C:\Program Files\IBM\WebSphere MQ\Errors).

To switch on trace for WebSphere MQ:

1. Start the WebSphere MQ Services program.
2. Right-click the Trace icon under **WebSphere MQ Services** and select **Trace Properties** from the context menu.
3. Click **Start** in the Trace Properties window (Figure 6-34), and click **OK**.


Trace Properties

Trace - MOOGLE

Current Status

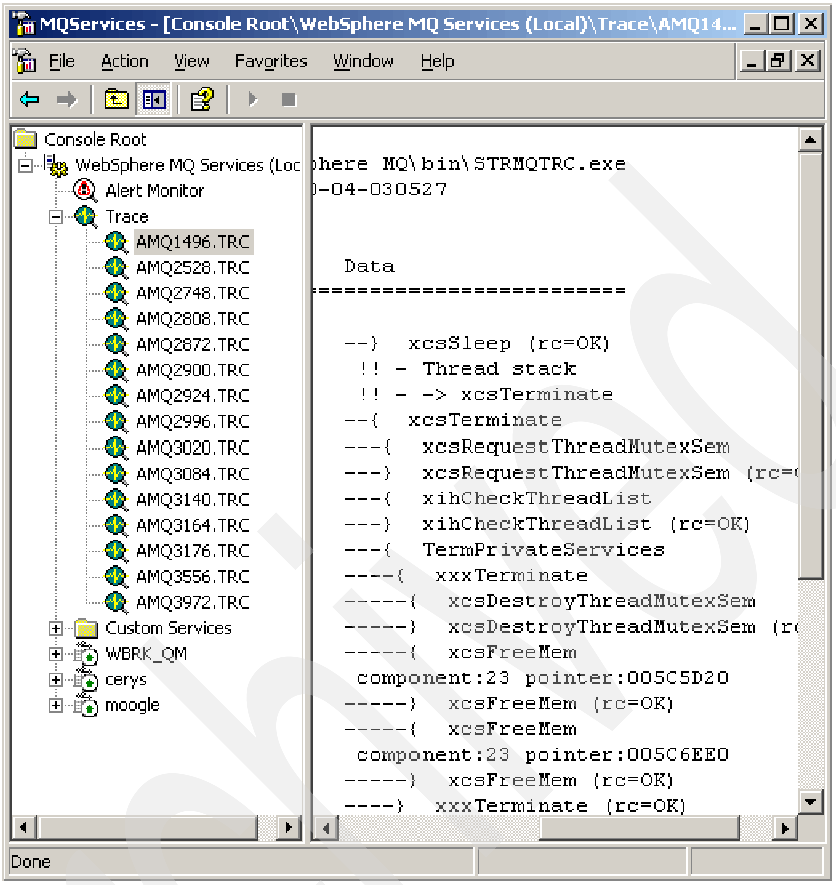
Status: Stopped

Startup

Type: Manual

*Figure 6-34 Starting Trace on WebSphere MQ*

4. To view the trace files that have been created, double-click the .TRC files under **Trace** in WebSphere MQ Services. Figure 6-35 on page 205 shows the trace as displayed.



```
phere MQ\bin\STRMQTRC.exe
)-04-030527


  Data
=========================

  --}  xcsSleep (rc=OK)
   !! - Thread stack
   !! - -> xcsTerminate
  --{  xcsTerminate
  ---{  xcsRequestThreadMutexSem
  ---}  xcsRequestThreadMutexSem (rc=(
  ---{  xihCheckThreadList
  ---}  xihCheckThreadList (rc=OK)
  ---{  TermPrivateServices
  ----{  xxxTerminate
  -----{  xcsDestroyThreadMutexSem
  -----}  xcsDestroyThreadMutexSem (r(
  -----{  xcsFreeMem
   component:23 pointer:005C5D20
  -----}  xcsFreeMem (rc=OK)
  -----{  xcsFreeMem
   component:23 pointer:005C6EE0
  -----}  xcsFreeMem (rc=OK)
  ----}  xxxTerminate (rc=OK)
```

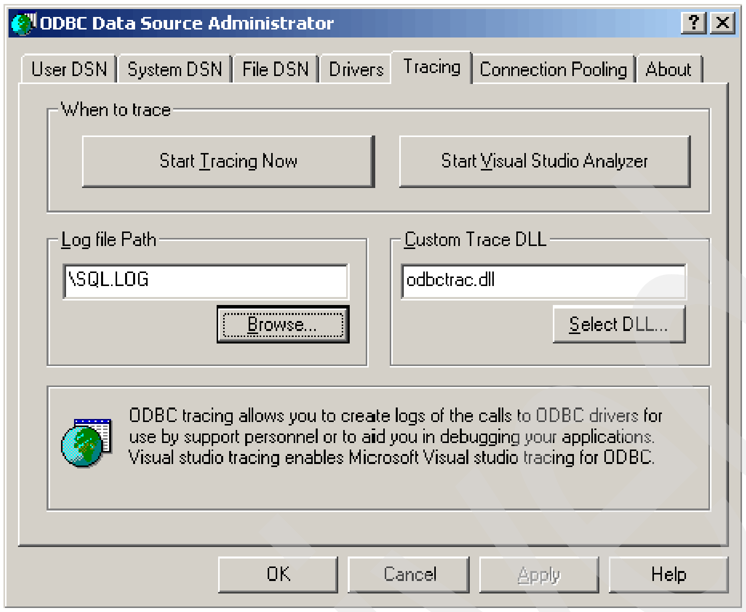
*Figure 6-35 Viewing trace files in WebSphere MQ*

Select **Stop** in Trace Properties to stop trace from being recorded on WebSphere MQ components and executables after the required trace has been generated.

## 6.3.6 ODBC trace

ODBC trace can be useful if problems related to database connections occur when using ESQL or WebSphere Business Integration Message Brokers components. To start ODBC tracing on Windows:

1. Open the **ODBC Data Source Administrator** program from the Windows Control Panel. (It is usually located under Administrative Tools, but the location depends on the version of Windows that is installed on the system.)
2. Select the **Tracing** tab in the ODBC Data Source Administrator (Figure 6-36 on page 206).


ODBC Data Source Administrator

User DSN | System DSN | File DSN | Drivers | Tracing | Connection Pooling | About

When to trace: Start Tracing Now | Start Visual Studio Analyzer

Log file Path: \SQL.LOG (Browse...)

Custom Trace DLL: odbctrac.dll (Select DLL...)

ODBC tracing allows you to create logs of the calls to ODBC drivers for use by support personnel or to aid you in debugging your applications. Visual studio tracing enables Microsoft Visual studio tracing for ODBC.

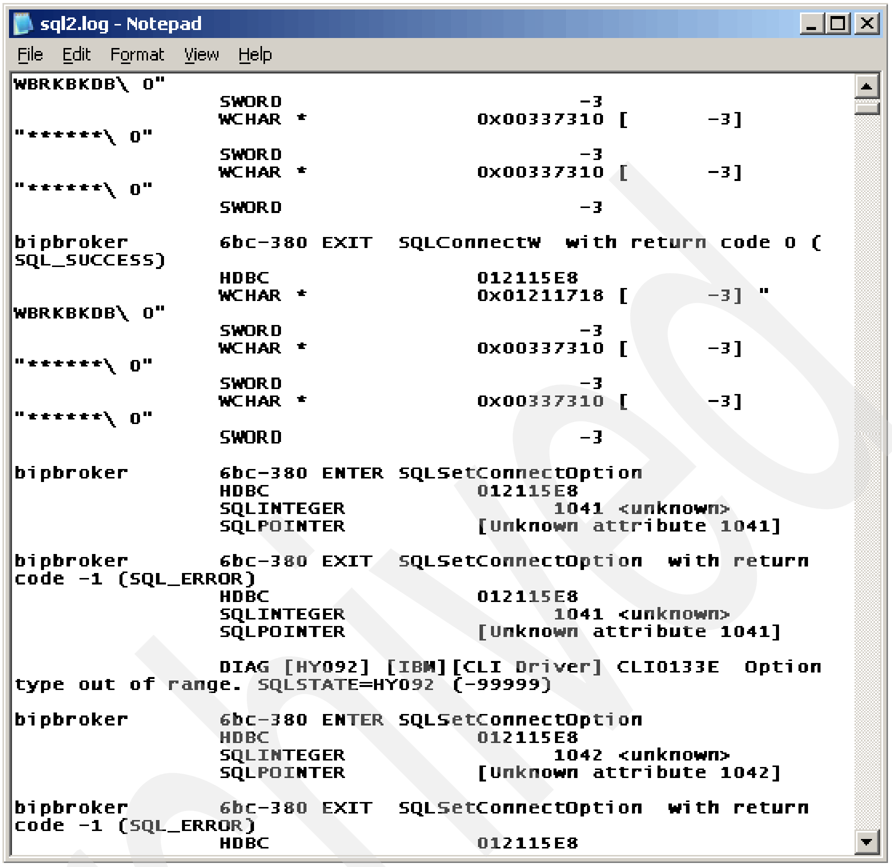
OK | Cancel | Apply | Help

*Figure 6-36 Starting ODBC trace*

3. Check the **Log file Path**. This can be changed to a different name and directory, if necessary.
4. Click **Start Tracing Now** to start tracing ODBC connections on the system.

Click **Stop Tracing Now** in the ODBC Data Source Administrator to stop collecting ODBC trace.

Figure 6-37 on page 207 shows an example of the output that is produced in the ODBC trace file. The success of an ODBC/SQL action is indicated by the return code, where 0 (zero) is successful and the value -1 is an error condition.



```
WBRKBKDB\ 0"
                SWORD                        -3
                WCHAR *             0x00337310 [       -3]
"******\ 0"
                SWORD                        -3
                WCHAR *             0x00337310 [       -3]
"******\ 0"
                SWORD                        -3

bipbroker       6bc-380 EXIT  SQLConnectW  with return code 0 (
SQL_SUCCESS)
                HDBC                012115E8
                WCHAR *             0x01211718 [       -3] "
WBRKBKDB\ 0"
                SWORD                        -3
                WCHAR *             0x00337310 [       -3]
"******\ 0"
                SWORD                        -3
                WCHAR *             0x00337310 [       -3]
"******\ 0"
                SWORD                        -3

bipbroker       6bc-380 ENTER SQLSetConnectOption
                HDBC                012115E8
                SQLINTEGER                1041 <unknown>
                SQLPOINTER          [Unknown attribute 1041]

bipbroker       6bc-380 EXIT  SQLSetConnectOption  with return
code -1 (SQL_ERROR)
                HDBC                012115E8
                SQLINTEGER                1041 <unknown>
                SQLPOINTER          [Unknown attribute 1041]

                DIAG [HY092] [IBM][CLI Driver] CLI0133E  Option
type out of range. SQLSTATE=HY092 (-99999)

bipbroker       6bc-380 ENTER SQLSetConnectOption
                HDBC                012115E8
                SQLINTEGER                1042 <unknown>
                SQLPOINTER          [Unknown attribute 1042]

bipbroker       6bc-380 EXIT  SQLSetConnectOption  with return
code -1 (SQL_ERROR)
                HDBC                012115E8
```

*Figure 6-37 Example contents of an ODBC trace log*

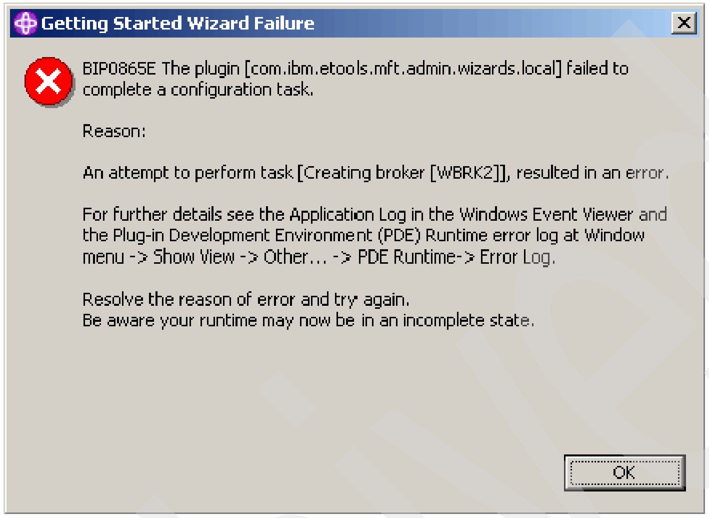
## 6.4 Troubleshooting common problems

This section describes some basic, common problems that may be experienced with WebSphere Business Integration Message Brokers and suggestions for overcoming these issues.

### 6.4.1 Getting Started wizard fails

The Getting Started Wizard may fail for a variety of reasons, but typically because of an underlying problem with the WebSphere Business Integration Message Brokers configuration that prevents the creation or starting of one of the components that the Getting Started Wizard creates.

If the Getting Started Wizard fails, the message shown in Figure 6-38 is displayed. This BIP error message indicates which task the wizard was performing when it failed and points the user to the PDE Runtime error log as a further source of information.



BIP0865E The plugin [com.ibm.etools.mft.admin.wizards.local] failed to complete a configuration task.

Reason:

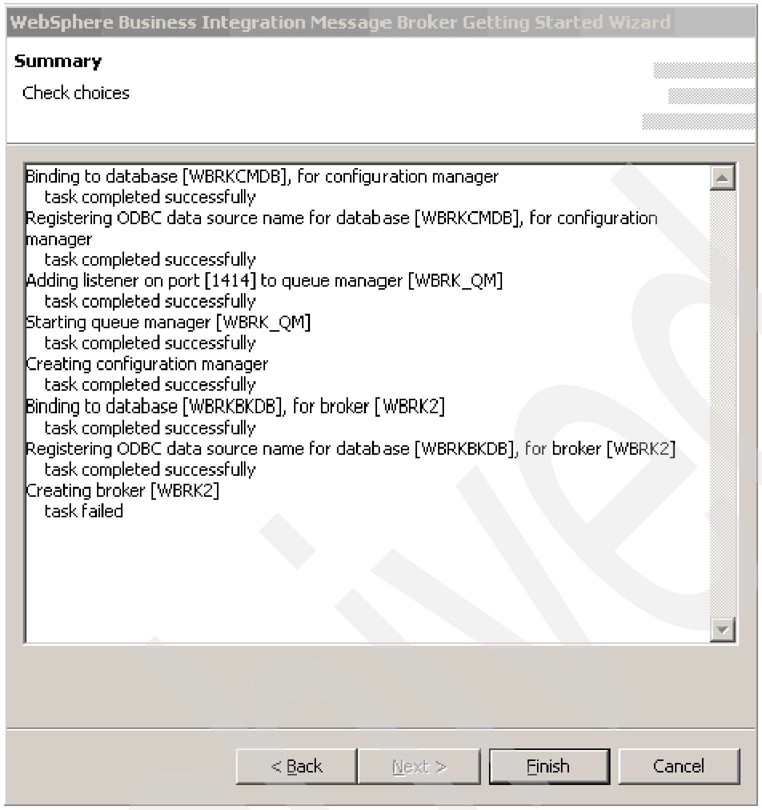
An attempt to perform task [Creating broker [WBRK2]], resulted in an error.

For further details see the Application Log in the Windows Event Viewer and the Plug-in Development Environment (PDE) Runtime error log at Window menu -> Show View -> Other... -> PDE Runtime-> Error Log.

Resolve the reason of error and try again.
Be aware your runtime may now be in an incomplete state.

*Figure 6-38 Error produced on Getting Started Wizard Failure*

The Summary page of the Getting Started wizard (Figure 6-39 on page 209) also indicates which task was being performed when the failure occurred.


WebSphere Business Integration Message Broker Getting Started Wizard
Summary
Check choices

```
Binding to database [WBRKCMDB], for configuration manager
   task completed successfully
Registering ODBC data source name for database [WBRKCMDB], for configuration
manager
   task completed successfully
Adding listener on port [1414] to queue manager [WBRK_QM]
   task completed successfully
Starting queue manager [WBRK_QM]
   task completed successfully
Creating configuration manager
   task completed successfully
Binding to database [WBRKBKDB], for broker [WBRK2]
   task completed successfully
Registering ODBC data source name for database [WBRKBKDB], for broker [WBRK2]
   task completed successfully
Creating broker [WBRK2]
   task failed
```

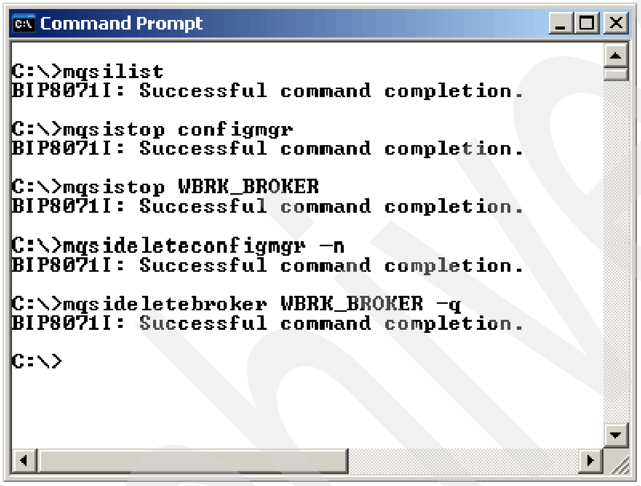
*Figure 6-39 Getting Started wizard Summary page*

As shown in Figure 6-19 on page 181, the PDE Runtime Error Log gives details of any BIP errors or other errors that were thrown during the running of the Getting Started wizard. This information may be enough to locate the cause of the problem. If the PDE Runtime Error Log does not provide enough information, then it is necessary to seek further information in the form of BIP or other error messages in the Application Log of the Windows Event Viewer.

The Getting Started wizard is designed to be run on a fresh system, after a WebSphere Business Integration Message Brokers installation, as the first stage in creating the basic configuration that is used to verify that install. It is not designed for creating components for production use. Problems can occur if the Getting Started Wizard is run on a system that already has components on it, and it is advisable to take the system back to a clean state before running the wizard.

As the error message in Figure 6-38 on page 208 indicates, when a failure occurs when running the Getting Started wizard, then the system it is run on can be left in an inconsistent state. One odd effect that often occurs is when an error occurs with the creation or starting of a component, it ends up partially created on a system.

Figure 6-40 shows the result of this kind of problem, running an **mqsilist** command after a failure of the Getting Started wizard. (This effect can happen, but is not seen with every type of failure.) No components are displayed on the system, so it appears that none were created. More often, the Configuration Manager is displayed as the result of the **mqsilist** command but the broker is missing.



```
C:\>mqsilist
BIP8071I: Successful command completion.

C:\>mqsistop configmgr
BIP8071I: Successful command completion.

C:\>mqsistop WBRK_BROKER
BIP8071I: Successful command completion.

C:\>mqsideleteconfigmgr -n
BIP8071I: Successful command completion.

C:\>mqsideletebroker WBRK_BROKER -q
BIP8071I: Successful command completion.

C:\>
```

*Figure 6-40 Cleaning up a system after Getting Started wizard failure*

The next two commands are used to stop the running components. If the components did not actually exist on the system, then these commands fail with a `Component does not exist` error message, which is different to the components existing but already being stopped. The same is true of the **mqsidelete** commands. These are used to clean the partially existing components off the system.

If such a problem is seen with the Getting Started wizard, after solving the original problem you should clean the system before running the wizard again. Refer to 6.4.6, "How to get to a clean configuration" on page 217 for assistance with this task.

## 6.4.2 Errors with the Message Brokers Toolkit

Persistent problems may occur with the Message Brokers Toolkit and the Eclipse Workbench: Java or NullPointer errors can occur in the PDE Runtime Error Log.

Another problem that may occur is that the perspectives and user settings in the Eclipse Workbench cannot be saved on shutdown of the workbench or be restored during startup of the workbench. Errors may also occur if the Eclipse Workbench is terminated unexpectedly. For any such errors that are not resolved through normal use of the Message Brokers Toolkit, the following method can be used to reset the Eclipse Workbench back to its original state at install time. This method, however, removes any user-defined perspectives and views.

1. Close the Message Brokers Toolkit.
2. Navigate to the Eclipse directory in the install path of WebSphere Business Integration Message Brokers (for example: C:\Program Files\IBM\WebSphere Business Integration Message Brokers\eclipse).
3. Open the Workspace directory from inside the Eclipse directory.
4. Select the .metadata directory and delete it.
5. Start the Message Brokers Toolkit. Figure 6-41 shows the message that displays on startup.

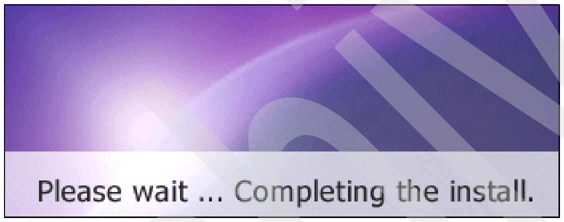
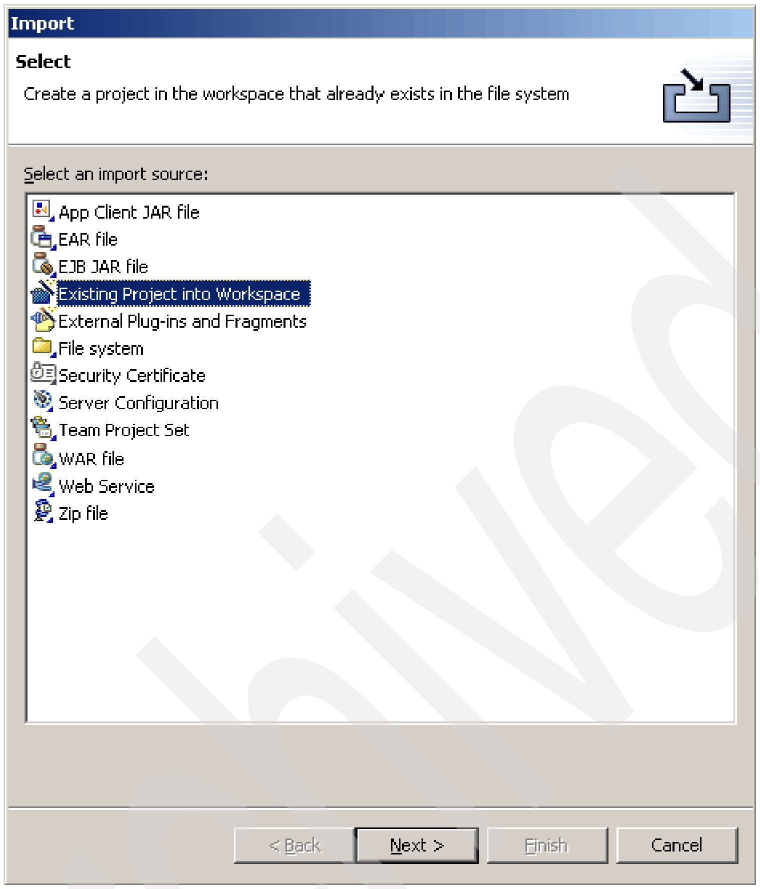

Please wait ... Completing the install.

*Figure 6-41 Eclipse startup message*

In order to display any existing projects, these must be reimported into the workspace. To do this:

6. Select **Import** from the **File** menu.
7. From the Import wizard (Figure 6-42 on page 212), select **Existing Project into Workspace** and click **Next**.


Import

Select

Create a project in the workspace that already exists in the file system

Select an import source:

- App Client JAR file
- EAR file
- EJB JAR file
- Existing Project into Workspace
- External Plug-ins and Fragments
- File system
- Security Certificate
- Server Configuration
- Team Project Set
- WAR file
- Web Service
- Zip file

*Figure 6-42 Importing existing projects into workspace*

8. Browse to the existing Workspace folder on the system (for example, C:\Program Files\IBM\WebSphere Business Integration Message Brokers\eclipse\workspace).
9. Select an existing project to import and click **OK**.
10. Click **Finish**.
11. Repeat steps 6 on page 211 through 10 for any existing projects, as only one project at a time can be imported to the workspace.

If these steps do not solve a particular problem that is occurring in the Message Brokers Toolkit, you should seek further assistance.

## 6.4.3 Problems connecting to the Configuration Manager

The Message Brokers Toolkit must communicate with the Configuration Manager in order to perform any types of deploy task. The properties for the connection to the Configuration Manager are set up in the Domain Connection wizard. The properties are stored in a file with a .configmgr extension and, if the properties change, they can be edited by right-clicking the Domain Connection.

Figure 6-43 shows the Domain Connection when the connection between the Message Brokers Toolkit and the Configuration Manager has not been established. The Domain Connection and the editor options beneath it are grayed out.

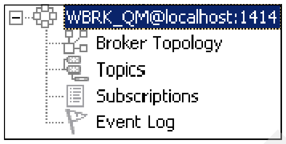
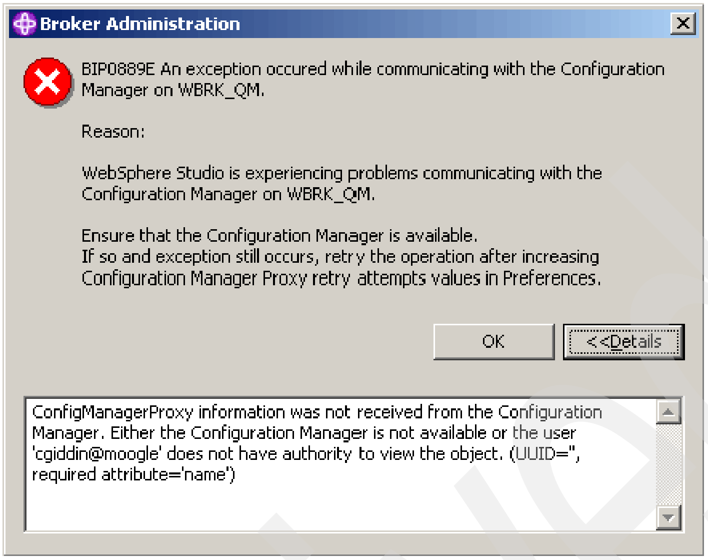


*Figure 6-43 Disconnected Broker Domain*

To establish the connection, right-click the Domain Connection and select **Connect** from the context menu. This makes the Message Brokers Toolkit attempt to connect to the Configuration Manager, and a progress bar is displayed. If the progress bar seems to stop or take a long time, there is likely to be a problem with the connection to the Configuration Manager.

If a failure occurs during connection to the Configuration Manager, then a BIP error message is displayed with the cause or possible causes of the problem visible in the Details section of the error message, as seen in Figure 6-44 on page 214. Several error messages may be displayed in order.



BIP0889E An exception occured while communicating with the Configuration Manager on WBRK_QM.

Reason:

WebSphere Studio is experiencing problems communicating with the Configuration Manager on WBRK_QM.

Ensure that the Configuration Manager is available.
If so and exception still occurs, retry the operation after increasing Configuration Manager Proxy retry attempts values in Preferences.

ConfigManagerProxy information was not received from the Configuration Manager. Either the Configuration Manager is not available or the user 'cgiddin@moogle' does not have authority to view the object. (UUID='', required attribute='name')

*Figure 6-44 Error message: communication problem with Configuration Manager*

Use the error information in the Details section of the BIP error message to help track down the cause of the problem. Some common solutions to problems with the Message Brokers Toolkit's connection to the Configuration Manager are:

- Confirm that the Configuration Manager is created.
- Confirm that the Configuration Manager is started sucessfully, as indicated by a BIP1003 message in the Application Log in the Windows Event Viewer.
- Confirm that the queue manager is available.
- Confirm that the queue manager that is specified in the Domain Connection configuration is correct.
- Confirm that the WebSphere MQ Listener is running.
- Confirm that nothing else is running on the same port as the WebSphere MQ Listener.
- Confirm that the WebSphere MQ Listener port that is specified in the Domain Connection configuration is correct.
- Ensure that the user that is running the Message Brokers Toolkit has the appropriate security authorities to access and run the Configuration Manager.

### 6.4.4 Problems with deployment

A variety of problems can occur with deployment actions, and this sections details a few of the common problems and solutions.

## Deleted brokers

If a broker is deleted on a system and then re-created with the same name, the broker will not be recognized by the Configuration Manager on the next deploy operation to it. This is because each broker and execution group on its first deploy is allocated a unique ID. This ID is referred to in subsequent deploys, and if the broker is re-created, the ID is no longer present.

If the broker is deleted and re-created without being deleted from the Configuration Manager, then BIP2062 and BIP2087 error messages are seen in the Message Brokers Toolkit Event Log in the next deploy operation.

To remedy this situation:

1. In the Broker Administration perspective in the Message Brokers Toolkit, right-click the re-created broker and select **Remove Deployed Children** from the context menu.
2. This displays successful configuration change messages in the Event Log (for example: BIP2056 and BIP4045). An error message from the Configuration Manager may also be present (BIP1536).
3. Deploy to the default execution group on the broker.

The execution group and broker are then assigned an ID and can be deployed to as normal.

If the Configuration Manager and its database have been re-created, then you must re-create any brokers in order to be able to deploy to them.

## Remote broker not responding

If a deploy to a remote broker fails or no response is received, these actions may help to solve or determine the cause of the problem.

- Confirm that the sender and receiver channels on the remote broker's queue manager are running.
- Confirm that the sender and receiver channels on the Configuration Manager's queue manager are running.
- Check the Event Log in the Message Brokers Toolkit for messages from the Configuration Manager or the broker.
- Check the Windows Event Viewer for error messages from the Configuration Manager or from WebSphere MQ.
- Check the local log on the broker system for error messages from the broker or from WebSphere MQ.

## Outstanding deploys

On occasion, an unprocessed deploy message can block other deploys from the Configuration Manager to a broker. This situation is resolved easily:

1. In the Broker Administration perspective in the Message Brokers Toolkit, right-click the Domain Connection.
2. Select **Cancel Deployment** from the context menu, as shown in Figure 6-45.

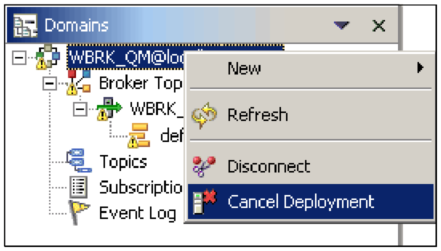


*Figure 6-45 Cancel Deployment option*

This clears any deployment messages from the Configuration Manager on the broker queues. WebSphere MQ channels must be running on any broker queue managers for any broker not on the same queue manager as the Configuration Manager.

## 6.4.5 Messages stuck on the input queue

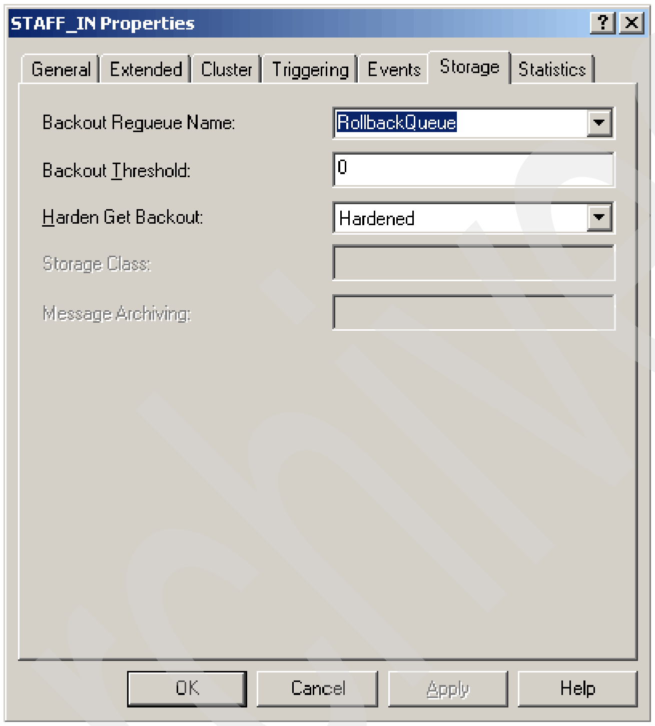
Messages may remain on the input queue of a message flow, usually for any of these reasons:

- Message flow is not running, perhaps because the message flow or broker is stopped or the deploy of the message flow failed. In this case, check the status of the message flow in the Message Brokers Toolkit, and the results of any deploys in the Event Log.
- There could also be issues such as a spelling mistake in the input node of the message flow, so the message flow is attempting to pick the messages up from a different queue. Any error or warning messages in the Application Log help to determine this kind of error.
- An error has occurred in the message flow and a message has been rolled back onto the input node. This message blocks any other messages that are put to the queue.

To resolve this problem, a dead letter queue can be defined for the input node:

1. Open the WebSphere MQ Explorer.

2. On the queue manager, create a new local queue and give this queue a descriptive name such as `RollbackQueue`.
3. Right-click the input queue and select **Properties** from the context menu.
4. Click the **Storage** tab in the queue properties.
5. Select **RollbackQueue** from the list of queues in the Backout Requeue Name (Figure 6-46), and click **OK**.



Backout Requeue Name: RollbackQueue
Backout Threshold: 0
Harden Get Backout: Hardened

*Figure 6-46 Setting a backout queue*

Any rolled messages on the input queue are now placed onto the backout queue instead.

## 6.4.6 How to get to a clean configuration

These instructions can be used to clean up a test or development system to a clean state. This list assumes that a default configuration has been created on the system.

1. Run **`mqsilist`** on a command line to determine the components that are present on the system.

2. Stop each of the components using the **mqsistop** command.
3. Delete the Configuration Manager using this command:

```
mqsideleteconfigmgr -n
```

4. Delete the brokers on the system using the -q parameter to delete the queue manager for the brokers. For example:

```
mqsideletebroker WBRK_BROKER -q
```

5. Delete the Configuration Manager's queue manager if it is not already deleted.
6. Delete any other components on the system (such as the User Name Server).
7. Navigate to the Eclipse directory in the WebSphere Business Integration Message Brokers (for example: C:\Program Files\IBM\WebSphere Business Integration Message Brokers\eclipse).
8. Open the Workspace directory and delete the .metadata directory to reset the Message Brokers Toolkit. Existing projects can be imported into the workspace using the instructions in 6.4.2, "Errors with the Message Brokers Toolkit" on page 210.

You also can drop the broker and Configuration Manager databases using the DB2 Control Center.

A

# Getting help

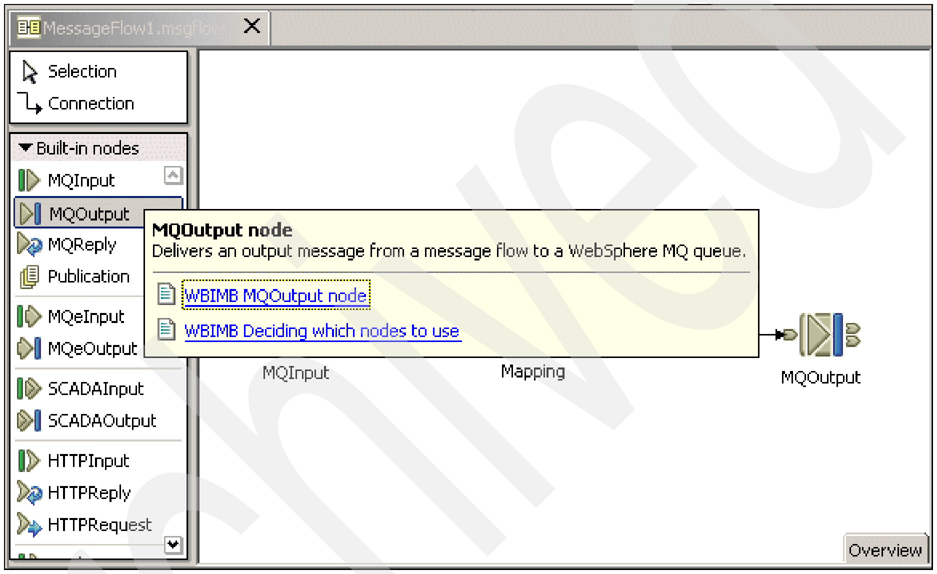
This appendix describes:

- Getting context-sensitive help within the Message Brokers Toolkit
- Viewing the product documentation in the help system
- Searching the help system
- Getting information from other sources
- Serving the help system from a central location
- Useful links

## Getting context-sensitive help

To get context-sensitive help from within the Message Brokers Toolkit:

1. Bring focus to any part of the user interface. For example, to view context-sensitive help, open the Message Flow Editor, then click the MQOutput node in the Node Palette.
2. Press F1. A yellow box, an infopop, is displayed. The infopop gives some high-level help, followed by a short list of links (Figure A-1).



**MQOutput node**
Delivers an output message from a message flow to a WebSphere MQ queue.

WBIMB MQOutput node

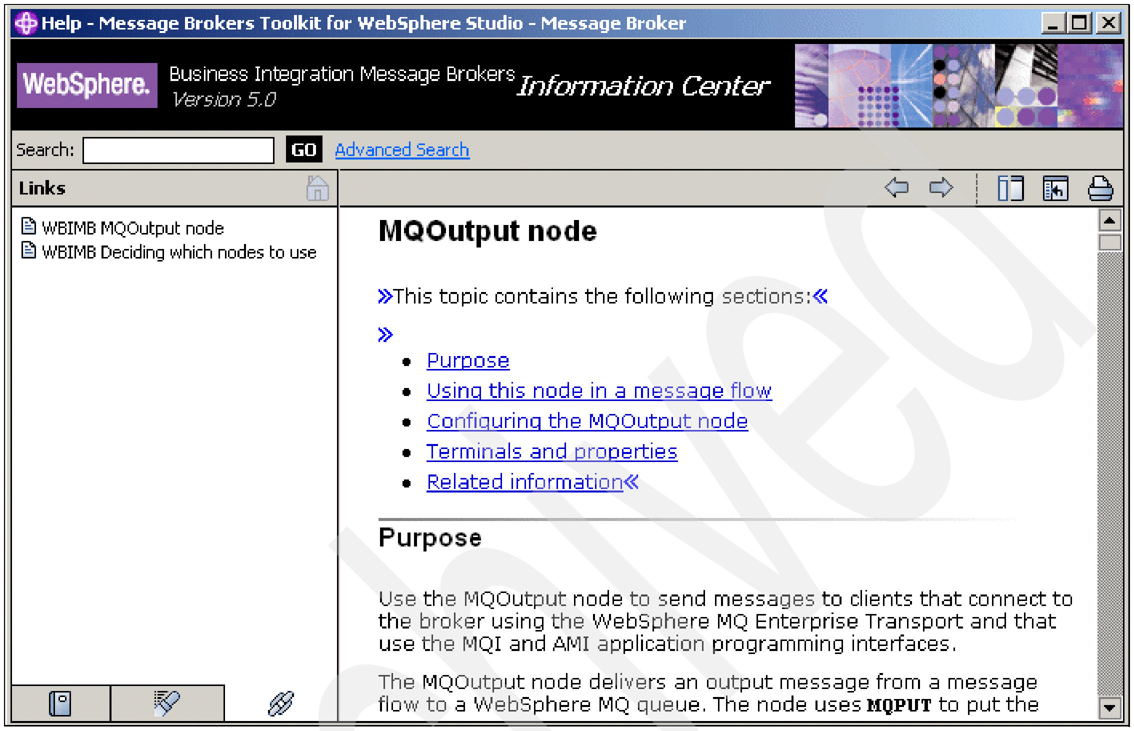
WBIMB Deciding which nodes to use

*Figure A-1 Infopop for MQOutput node in the Node Palette*

3. If you need more information, click one of the links to open the help system at the relevant page of the product documentation.

When you click one of the links in an infopop, the help system for the Message Brokers Toolkit opens at the relevant topic. If you click the **WBIMB MQOutput node** link in the example, the help system opens the WebSphere Business Integration Message Brokers product documentation at the topic titled `MQOutput node`.

The other links in the infopop are listed in the Links view, on the left side of the help system window (Figure A-2). Click each link to find additional, relevant information in the help system.


Help - Message Brokers Toolkit for WebSphere Studio - Message Broker
WebSphere. Business Integration Message Brokers Version 5.0 Information Center
Search: GO Advanced Search
Links
WBIMB MQOutput node
WBIMB Deciding which nodes to use

MQOutput node

»This topic contains the following sections:«

»
- Purpose
- Using this node in a message flow
- Configuring the MQOutput node
- Terminals and properties
- Related information«

Purpose

Use the MQOutput node to send messages to clients that connect to the broker using the WebSphere MQ Enterprise Transport and that use the MQI and AMI application programming interfaces.

The MQOutput node delivers an output message from a message flow to a WebSphere MQ queue. The node uses **MQPUT** to put the

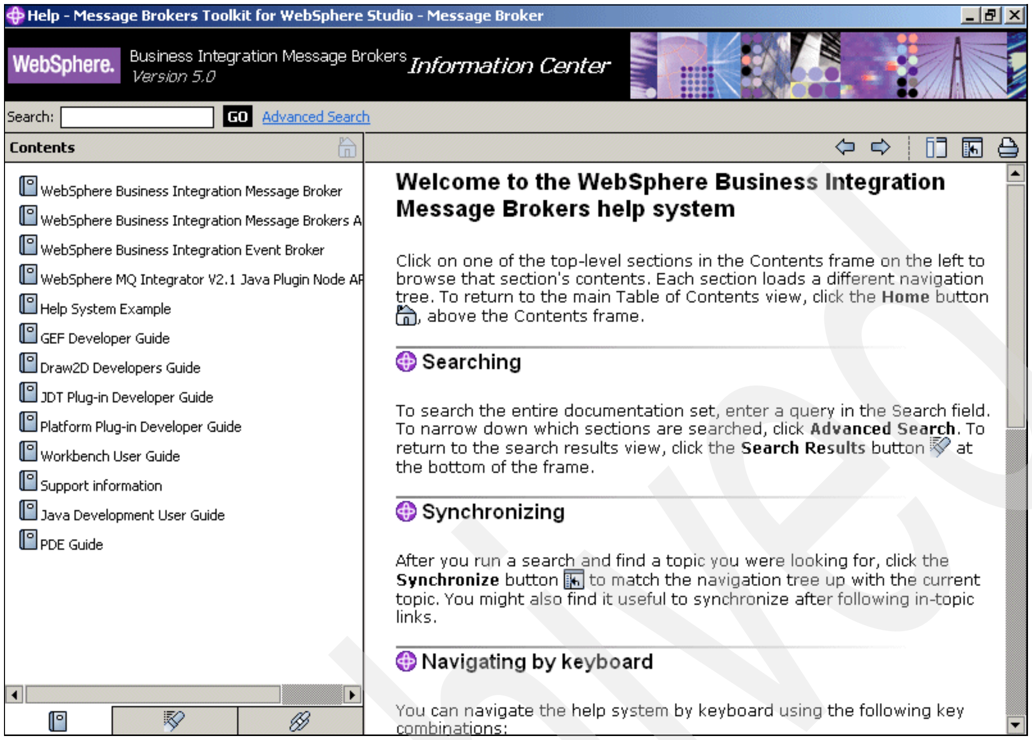
*Figure A-2 Infopop links in the help system*

## Getting help from the product documentation

All of the product documentation for WebSphere Business Integration Message Broker and WebSphere Business Integration Event Broker is in the Message Brokers Toolkit help system.

To open the help system, you can:

- Click a link in an infopop to open the help system at the relevant page.
- Click **Help** → **Help Contents**. The help system opens at the help system Welcome page (Figure A-3 on page 222).


Help - Message Brokers Toolkit for WebSphere Studio - Message Broker
WebSphere.
Business Integration Message Brokers
Version 5.0
Information Center
Search:
GO
Advanced Search
Contents
WebSphere Business Integration Message Broker
WebSphere Business Integration Message Brokers A
WebSphere Business Integration Event Broker
WebSphere MQ Integrator V2.1 Java Plugin Node AF
Help System Example
GEF Developer Guide
Draw2D Developers Guide
JDT Plug-in Developer Guide
Platform Plug-in Developer Guide
Workbench User Guide
Support information
Java Development User Guide
PDE Guide

**Welcome to the WebSphere Business Integration Message Brokers help system**

Click on one of the top-level sections in the Contents frame on the left to browse that section's contents. Each section loads a different navigation tree. To return to the main Table of Contents view, click the **Home** button, above the Contents frame.

**Searching**

To search the entire documentation set, enter a query in the Search field. To narrow down which sections are searched, click **Advanced Search**. To return to the search results view, click the **Search Results** button at the bottom of the frame.

**Synchronizing**

After you run a search and find a topic you were looking for, click the **Synchronize** button to match the navigation tree up with the current topic. You might also find it useful to synchronize after following in-topic links.

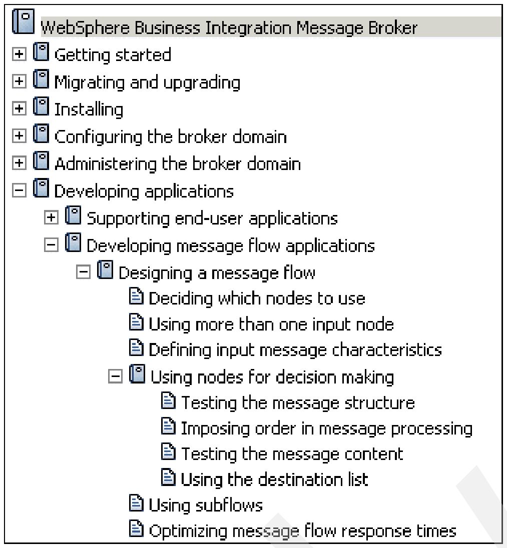
**Navigating by keyboard**

You can navigate the help system by keyboard using the following key combinations:

*Figure A-3 Help system welcome page*

The Contents view on this page lists subjects on which help is available. The product documentation for Message Broker is the first item in the Contents list, and the product documentation for Event Broker is the third item in the Contents list. The description assumes that you are working with the Message Broker version of the product, but the Event Broker product documentation is similar in structure and content.

The WebSphere Business Integration Message Brokers help system is organized by the tasks that you are likely to want to perform, such as configuring and administering a system and diagnosing problems, or developing, debugging, and deploying applications. The first eight sections of the Message Brokers help system cover tasks such as this, first at a high level then, as you expand the sections, going into more detail (Figure A-4 on page 223).



- WebSphere Business Integration Message Broker
  - Getting started
  - Migrating and upgrading
  - Installing
  - Configuring the broker domain
  - Administering the broker domain
  - Developing applications
    - Supporting end-user applications
    - Developing message flow applications
      - Designing a message flow
        - Deciding which nodes to use
        - Using more than one input node
        - Defining input message characteristics
        - Using nodes for decision making
          - Testing the message structure
          - Imposing order in message processing
          - Testing the message content
          - Using the destination list
        - Using subflows
        - Optimizing message flow response times

*Figure A-4 Expanding the navigation tree for detailed task information*

In addition to the sections that provide help with performing tasks, two sections are dedicated to providing conceptual information, such as defining "broker" and "Configuration Manager," and reference material, such as a list of all commands for Message Brokers on distributed platforms.

After the Concepts and Reference sections are:

- ► The Samples Gallery, which contains documentation for the sample applications that are supplied with Message Brokers.
- ► The Glossary of terms used in Message Brokers.
- ► The Diagnostic messages tool into which you can type the number of an error or warning message to find out more information.

If you are new to WebSphere Business Integration Message Brokers, get a product overview and experiment with some simple tasks that are detailed in the Getting Started section of the help system. Explore the Message Brokers and the Message Brokers Toolkit by preparing and running some of the sample applications that are documented in the Samples Gallery.

You might also find useful the Eclipse Workbench User Guide, which describes how to work with the basic framework in which the Message Brokers Toolkit is built. For example, the Workbench User Guide describes how to customize the workbench, such as rearranging the views within perspectives.

## Searching for information in the help system

An alternative to navigating the help system is to use the Search facility to find information about a specific subject. The Search field (Figure A-5) is located near the top of the help system window.

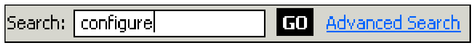


*Figure A-5 The Search field*

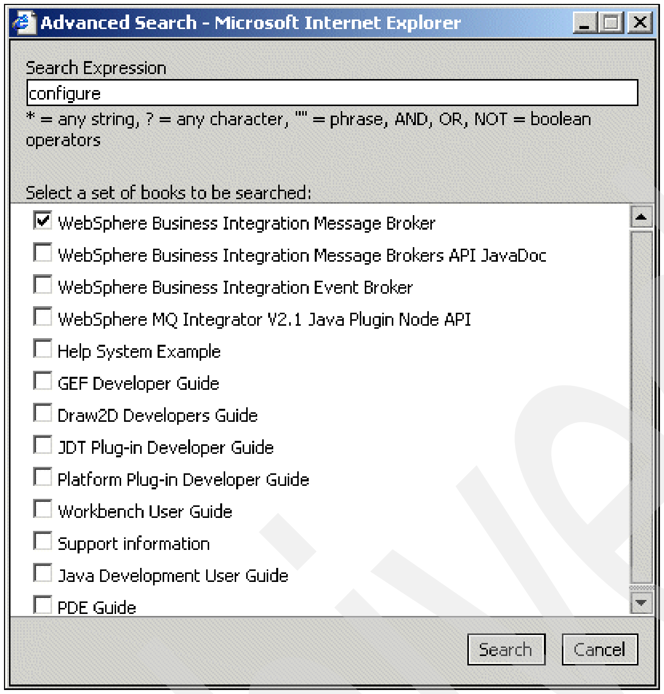
To search for a piece of information in the help system:

1. In the Search field, type one or more words, for example, `configure`.
2. Click **Go** (or press Enter). The search results are listed, in order of relevance, on the left side of the help system window. The percentage match of each search result is shown, where 100% means that the result is most likely to be the topic that you want.

The first time that you perform a search after installing the documentation, the documentation plug-ins are indexed. This can take a few minutes but will not repeat for subsequent searches.

### Narrowing the scope of the search

By default, the Search facility searches the information about all of the subjects that are listed in the Contents view. To narrow the scope of the search to information about specific subjects, click **Advanced Search** (Figure A-6 on page 225).



Search Expression

configure

* = any string, ? = any character, "" = phrase, AND, OR, NOT = boolean operators

Select a set of books to be searched:

- [✓] WebSphere Business Integration Message Broker
- [ ] WebSphere Business Integration Message Brokers API JavaDoc
- [ ] WebSphere Business Integration Event Broker
- [ ] WebSphere MQ Integrator V2.1 Java Plugin Node API
- [ ] Help System Example
- [ ] GEF Developer Guide
- [ ] Draw2D Developers Guide
- [ ] JDT Plug-in Developer Guide
- [ ] Platform Plug-in Developer Guide
- [ ] Workbench User Guide
- [ ] Support information
- [ ] Java Development User Guide
- [ ] PDE Guide

*Figure A-6 The Advanced Search dialog box*

To search a subset of the books in Advanced Search:

1. Clear the check box beside each subject that you do not want to search.
2. In the Search Expression field, type one or more words and click **Search**. The Advanced Search dialog box closes and the results appear in the help system window.

Be aware that when you clear the search field, the scope of the search reverts to the whole help system again.

## Searching on more than one word

If you type more than one word in the Search field, the results consist only of topics that contain all of the words in the content. For example, if you enter the words `configure` and `broker`, the search results list only topics that contain both `configure` and `broker` in their content, such as `Configuring a broker domain in the workbench`. The Boolean `AND` operator between the words is implied unless you use another operator, such as `OR`.

See the Workbench User Guide (**Workbench User Guide** → **Tasks** → **Using the help system** → **Searching online help**) in the help system for more information about using operators and wildcards in an Eclipse-based help system.

## Orienting yourself in the help system

When you click an item in the search results, or when you follow a link from one topic to another within the help system, the navigation tree does not automatically change to indicate your new location. This is to prevent the navigation tree from constantly refreshing while you move between help topics.

To synchronize the navigation tree with the topic that you are viewing, click the Synchronize Navigation button at the top-right of the help system window (the second icon from the right in Figure A-7).

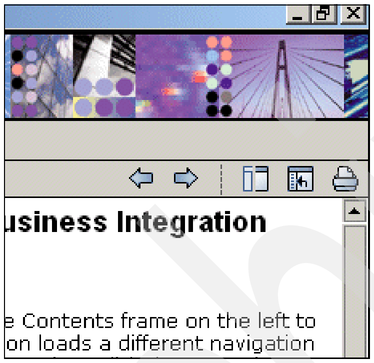

usiness Integration
e Contents frame on the left to
on loads a different navigation

*Figure A-7 Help system navigation buttons: Go Back, Go Forward, Toggle Navigation, Synchronize Navigation, Print*

## Updating the help system and infopops

Appendix B, "Updating the product" on page 229 describes how to search for product updates on the Web using Update Manager. Updates to the help system and infopops are also available via Update Manager.

## Getting help from other sources

Support Information, in the Contents list of the help system, contains an overview of the resources that are available to help solve problems that you might have when using WebSphere Business Integration Message Brokers. It also suggests the type of information that you should collect to help IBM software specialists diagnose and fix problems.

In addition, Support Information provides links to Message Brokers newsgroups and Web sites, and a search interface to help you find information in the Message Brokers support documents on the Web.

## Serving the help system from a single location

You might want to consider serving the WebSphere Business Integration Message Brokers help system from a central location within your organization if, for example:

- A large number of people within the organization need regular access to the help system and to updates for the help system.
- Many of the people within the organization who regularly use the help system do not have Internet access and so cannot easily obtain and install updates.

By installing the help WebSphere Studio workbench and the WebSphere Business Integration Message Brokers help system on a server, anyone with access to that server can view the help system using an ordinary Web browser. The main benefit is that updates to the product documentation can be applied to a single instance of the help system so that everyone in the organization is viewing up-to-date information.

Be aware, though, that if users are accessing the help from a central server, links in the documentation that launch graphical interface actions (such as starting a wizard) no longer work.

For instructions for installing and configuring the help system as a centralized infocenter, see the Platform Plug-in Developer Guide in the help system: Expand **Platform Plug-in Developer Guide** → **Reference** → **Other Reference Information** → **Installing the help system as an infocenter**.

## Useful links

Download fix packs for WebSphere Business Integration Message Brokers

http://www-306.ibm.com/software/integration/mqfamily/support/summary/wbib.html

View the latest readme file for WebSphere Business Integration Message Brokers

http://www.ibm.com/software/integration/mqfamily/support/readme/

Download the latest Install Guide for WebSphere Business Integration Message Brokers

`ftp://ftp.software.ibm.com/software/integration/wbibrokers/docs/`

View the latest list of software prerequisites

`http://www-306.ibm.com/software/integration/wbimessagebroker/requirements/`

Download fix packs for WebSphere MQ Family products

`http://www-306.ibm.com/software/integration/mqfamily/support/summary/index.html`

Download Service Packs for Microsoft Windows

`http://www.windowsupdate.com`

Download FixPaks for DB2

`http://www-306.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report`

Download a standalone version of the help system

`ftp://ftp.software.ibm.com/software/integration/wbibrokers/docs/wbimb_help.zip`

Download the WebSphere MQ Family manuals in PDF format

`http://www-306.ibm.com/software/integration/mqfamily/library/manualsa/manuals/crosslatest.html`

B

# Updating the product

This appendix describes:

- Searching for new product updates using Update Manager
- Installing updates of the product and documentation
- Checking what features are installed

## Staying up-to-date with Update Manager

Update Manager is a feature of Eclipse, the workbench on which Message Brokers Toolkit for WebSphere Studio is built. Update Manager searches in specified locations for new Message Brokers Toolkits, including documentation and updates. The national language versions of the documentation are also published on an Update Manager Web site.

To use Update Manager, you must have:

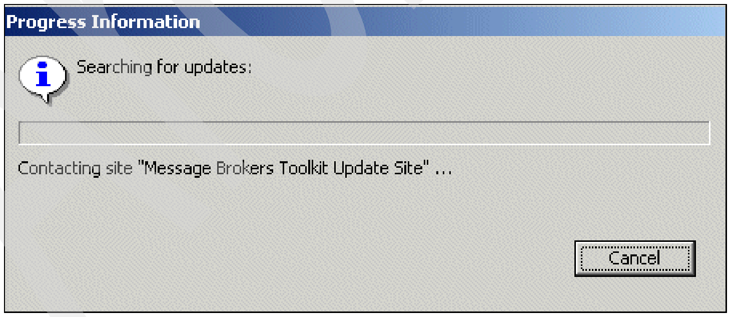
- A full installation of WebSphere Business Integration Message Broker or WebSphere Business Integration Event Broker, including the Message Brokers Toolkit
- An Internet connection

The Update Manager is a separate perspective within the Message Brokers Toolkit. Update Manager maintains a record of all features that you installed when you installed Message Brokers Toolkit, including any updates that you have installed using Update Manager since the initial product installation.

## Searching for new product updates

The Message Brokers Toolkit has all of the information that it needs to search for new product and documentation updates. To search for new updates:

1. In the Message Brokers Toolkit, click **Help** → **Software Updates** → **New Updates**. A progress bar is displayed as Update Manager contacts the Message Brokers Toolkit update site to check for new updates (Figure B-1).



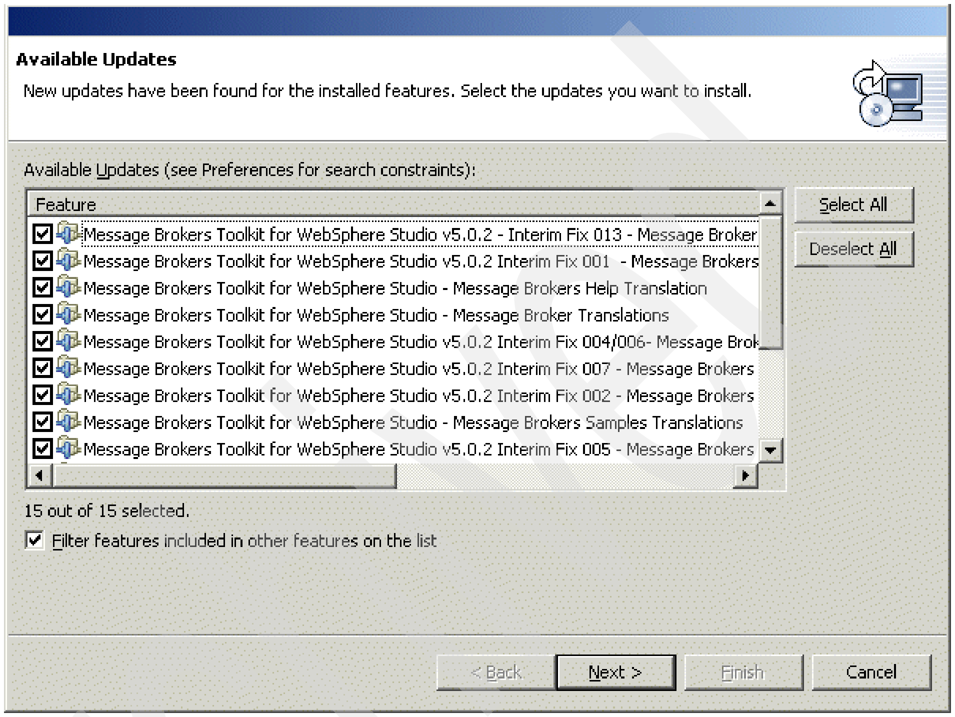
Searching for updates:

Contacting site "Message Brokers Toolkit Update Site" ...

*Figure B-1 Searching for new updates on Message Brokers Toolkit update site*

2. The Update Manager retrieves a list of the new updates that are available for your installation of Message Brokers Toolkit. This step could take a few minutes to complete.

3. When Update Manager displays all of the available new updates (Figure B-2), select the features that you want to install, then click **Next**. If you are not sure what you need to install, click **Select All** to install all of the new updates that are available.



**Available Updates**

New updates have been found for the installed features. Select the updates you want to install.

Available Updates (see Preferences for search constraints):

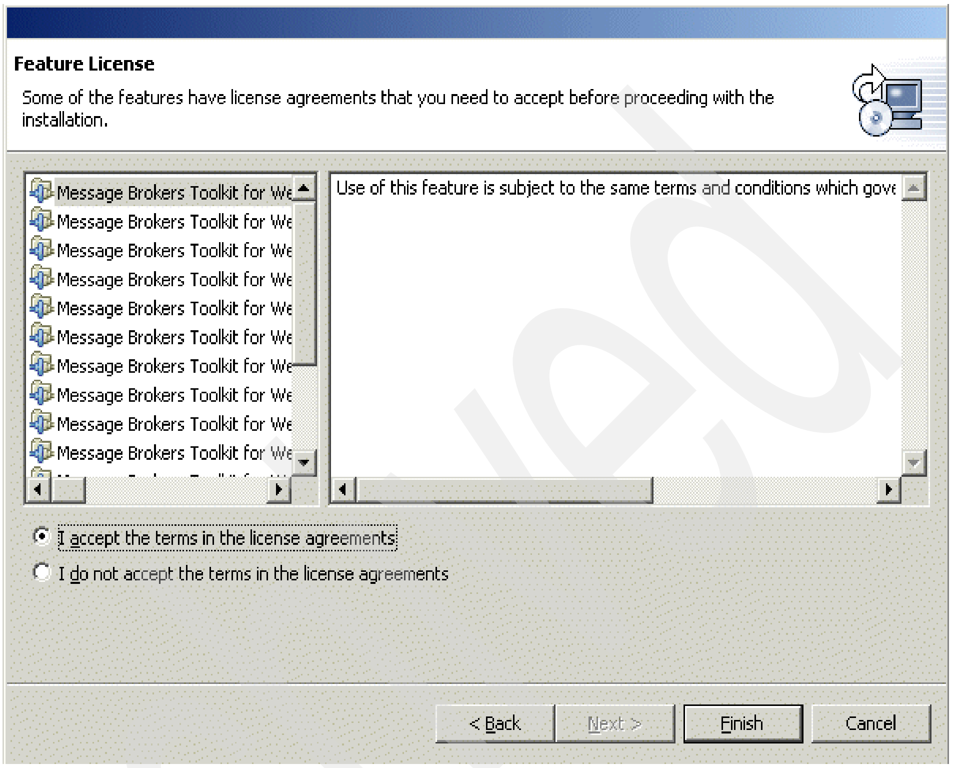
| Feature |
|---|
| ☑ Message Brokers Toolkit for WebSphere Studio v5.0.2 - Interim Fix 013 - Message Broker |
| ☑ Message Brokers Toolkit for WebSphere Studio v5.0.2 Interim Fix 001 - Message Brokers |
| ☑ Message Brokers Toolkit for WebSphere Studio - Message Brokers Help Translation |
| ☑ Message Brokers Toolkit for WebSphere Studio - Message Broker Translations |
| ☑ Message Brokers Toolkit for WebSphere Studio v5.0.2 Interim Fix 004/006- Message Brok |
| ☑ Message Brokers Toolkit for WebSphere Studio v5.0.2 Interim Fix 007 - Message Brokers |
| ☑ Message Brokers Toolkit for WebSphere Studio v5.0.2 Interim Fix 002 - Message Brokers |
| ☑ Message Brokers Toolkit for WebSphere Studio - Message Brokers Samples Translations |
| ☑ Message Brokers Toolkit for WebSphere Studio v5.0.2 Interim Fix 005 - Message Brokers |

15 out of 15 selected.

☑ Filter features included in other features on the list

*Figure B-2 The list of available updates*

4. When you are prompted, accept the terms in the license agreement and click **Next** (Figure B-3).


Feature License

Some of the features have license agreements that you need to accept before proceeding with the installation.

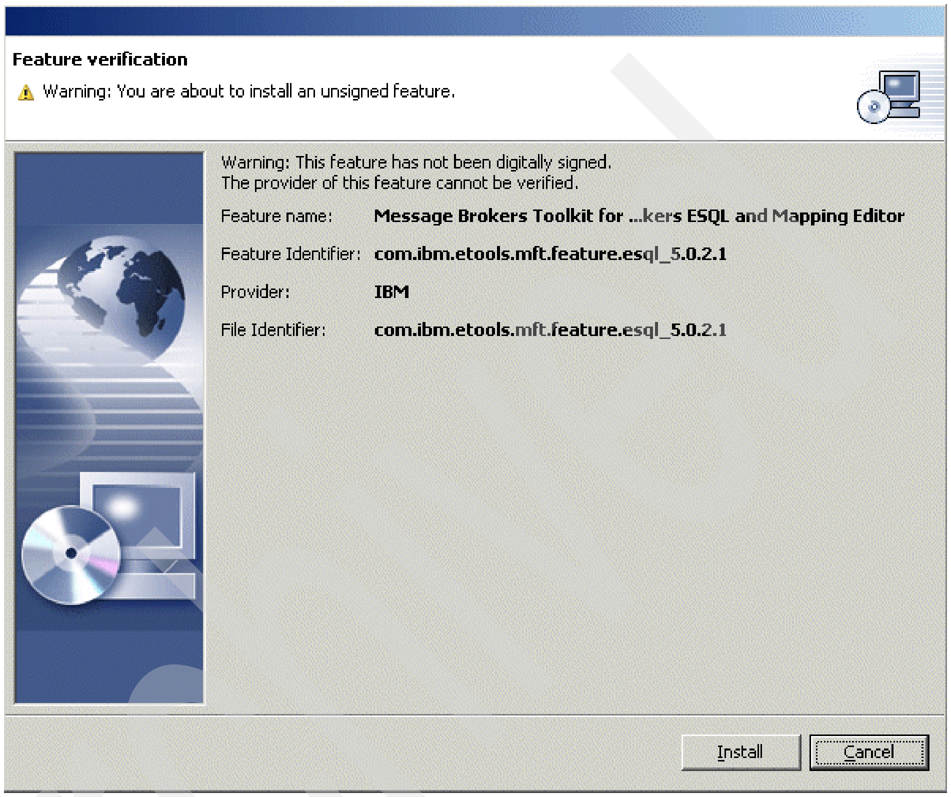
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We

Use of this feature is subject to the same terms and conditions which gove

[✓ I accept the terms in the license agreements]
I do not accept the terms in the license agreements

*Figure B-3 The update package license agreement*

5. When a warning is displayed saying that the feature that you are installing is not digitally signed, if you are sure that the feature is from IBM, click **Install** (Figure B-4).


Feature verification

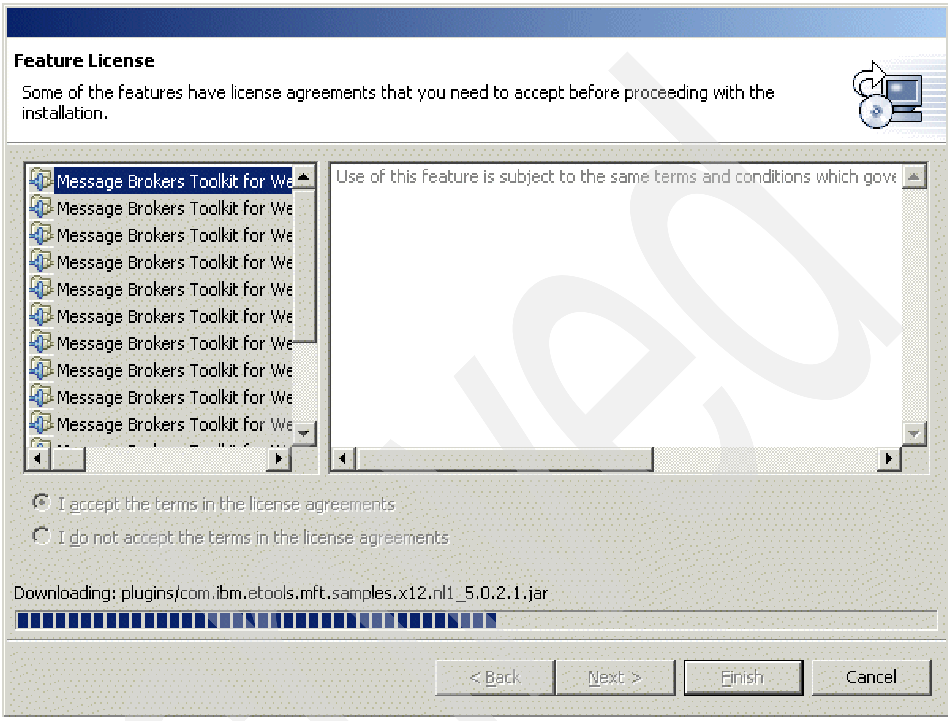
Warning: You are about to install an unsigned feature.

Warning: This feature has not been digitally signed.
The provider of this feature cannot be verified.

| | |
|---|---|
| Feature name: | Message Brokers Toolkit for ...kers ESQL and Mapping Editor |
| Feature Identifier: | com.ibm.etools.mft.feature.esql_5.0.2.1 |
| Provider: | IBM |
| File Identifier: | com.ibm.etools.mft.feature.esql_5.0.2.1 |

*Figure B-4 Update Manager: Warning that the feature is not digitally signed*

6. The installation continues (Figure B-5). A warning is displayed for each feature that you install. If you are sure that the feature is from IBM, click **Install**.


**Feature License**

Some of the features have license agreements that you need to accept before proceeding with the installation.

Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We
Message Brokers Toolkit for We

Use of this feature is subject to the same terms and conditions which gove

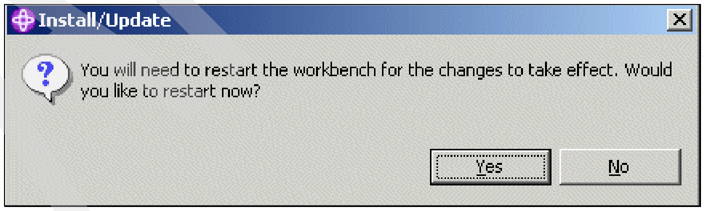
I accept the terms in the license agreements

I do not accept the terms in the license agreements

Downloading: plugins/com.ibm.etools.mft.samples.x12.nl1_5.0.2.1.jar

*Figure B-5 Update Manager: Installing a feature*

7. When all updates have been installed, click **Yes** to restart the Message Brokers Toolkit. When the Message Brokers Toolkit restarts, the installation is complete.


Install/Update

You will need to restart the workbench for the changes to take effect. Would you like to restart now?

*Figure B-6 Update Manger: restarting the Message Brokers Toolkit*

8. Repeat step 1 on page 230 through step 6 when you want to check for new updates to the Message Brokers Toolkit or to the product documentation.

# Abbreviations and acronyms

| | |
|---|---|
| **CWF** | Custom Wire Format |
| **ESQL** | Extended Structured Query Language |
| **IBM** | International Business Machines Corporation |
| **ITSO** | International Technical Support Organization |
| **JDBC** | Java database connectivity |
| **JRE** | Java Runtime Environment |
| **MDAC** | Microsoft Data Access Component |
| **ODBC** | Open database connectivity |
| **RAD** | Rapid Application Development |
| **SQL** | Structured Query Language |
| **TDS** | Tagged Delimited String |
| **XML** | Extensible Markup Language |

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks

For information about ordering this publication, see “How to get IBM Redbooks” on page 238.

- *WebSphere Business Integration Pub/Sub Solutions*, SG24-6088

## Online resources

These Web sites and URLs are also relevant as further information sources:

Download fix packs for WebSphere Business Integration Message Brokers

`http://www-306.ibm.com/software/integration/mqfamily/support/summary/wbib.html`

View the latest Readme for WebSphere Business Integration Message Brokers

`http://www.ibm.com/software/integration/mqfamily/support/readme/`

Download the latest Install Guide for WebSphere Business Integration Message Brokers

`ftp://ftp.software.ibm.com/software/integration/wbibrokers/docs/`

View the latest list of software prerequisites

`http://www-306.ibm.com/software/integration/wbimessagebroker/requirements/`

Download fix packs for WebSphere MQ Family products

`http://www-306.ibm.com/software/integration/mqfamily/support/summary/index.html`

Download Service Packs for Microsoft Windows

`http://www.windowsupdate.com`

Download FixPaks for DB2

`http://www-306.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report`

Download a standalone version of the help system

ftp://ftp.software.ibm.com/software/integration/wbibrokers/docs/wbimb_help.zip

Download the WebSphere MQ Family manuals in PDF format

http://www-306.ibm.com/software/integration/mqfamily/library/manualsa/manuals/crosslatest.html

## How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications, and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

**ibm.com**/redbooks

## Help from IBM

IBM Support and downloads

**ibm.com**/support

IBM Global Services

**ibm.com**/services

# Index

WebSphere Business Integration Message Broker Basics

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

# WebSphere Business Integration Message Broker Basics

**Introduces WebSphere Business Integration Message Broker**

**Describes basic installation and configuration tasks**

**Explores the Message Brokers Toolkit**

This IBM Redbook provides an overview of the latest release of WebSphere Business Integration Message Brokers and the new Message Brokers Toolkit for WebSphere Studio.

It covers the following topics:
- Installing WebSphere Business Integration Message Broker and WebSphere Business Integration Event Broker on Windows
- Creating and verifying a default configuration
- Developing a simple message flow and message set in the Message Brokers Toolkit
- Deploying a message flow application to a broker
- Diagnosing and fixing problems

The book also describes where to find more information, including product documentation and sample applications.

SG24-7090-00 ISBN 0738497851