WebSphere Event Broker

Migration

Version 6 Release 0

WebSphere Event Broker

Migration

Version 6 Release 0

Contents

About this topic collection

This PDF has been created from the WebSphere Event Broker Version 6.0 (Fix Pack 7 update, March 2008) information center topics. Always refer to the WebSphere Event Broker online information center to access the most current information. The information center is periodically updated on the document update site and this PDF and others that you can download from that Web site might not contain the most current information.

The topic content included in the PDF does not include the "Related Links" sections provided in the online topics. Links within the topic content itself are included, but are active only if they link to another topic in the same PDF collection. Links to topics outside this topic collection are also shown, but these attempt to link to a PDF that is called after the topic identifier (for example, ac12340_.pdf) and therefore fail. Use the online information to navigate freely between topics.

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

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

Part 1. Migrating

Migrating and upgrading

To migrate a broker domain to WebSphere® Event Broker Version 6.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, and then complete post-migration tasks.

Migration information is regularly updated on the WebSphere Event Broker support Web page with the latest details available. Look under the heading ″Solve a problem″ for the document ″Problems and solutions when migrating″.

The tasks in this section refer to installing Version 6.0; see the Installation Guide for details of how to complete this task.

You can migrate to WebSphere Event Broker Version 6.0 from the following products:
- WebSphere MQ Event Broker Version 2.1
- WebSphere Business Integration Event Broker Version 5.0

If you have applied CSD02, CSD03, CSD04, or CSD05 to Version 2.1, you must apply an additional APAR (IY45459) in order for migration to be successful. APAR IY45459 is included in CSD06.

The instructions apply to all operating systems supported by WebSphere MQ Event Broker Version 2.1 and WebSphere Business Integration Event Broker Version 5.0.

Version 5.1 Message Brokers Toolkit: if you are using the Version 5.1 Message Brokers Toolkit, follow the instructions in this section, replacing all references to ″Version 5.0 Message Brokers Toolkit″ with ″Version 5.1 Message Brokers Toolkit″.

If you are already using WebSphere MQ Publish/Subscribe, migrate your applications to use the publish/subscribe functions provided by WebSphere Event Broker Version 6.0 following the instructions in this section.

This section contains the following topics:
- "Coexistence with previous versions and other products"
- "Migrating from WebSphere MQ Event Broker Version 2.1 to WebSphere Event Broker Version 6.0" on page 12
- "Migrating from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0" on page 17
- "Restoring migrated components to previous versions" on page 40
- Migrating WebSphere MQ Publish/Subscribe applications

  WebSphere Event Broker support Web page

Coexistence with previous versions and other products

WebSphere Event Broker Version 6.0 can coexist and interoperate with previous versions and with other products.

This topic contains advice for the following scenarios:

- "Coexistence with previous versions of the product installed on the same computer"
- "Coexistence of Version 6.0 components with components from previous versions" on page 5
- "Coexistence with other products" on page 6

Coexistence with previous versions of the product installed on the same computer

WebSphere Event Broker Version 6.0 can coexist with either WebSphere MQ Event Broker Version 2.1 or WebSphere Business Integration Event Broker Version 5.0 on the same computer.

- You cannot install both WebSphere MQ Event Broker Version 2.1 and WebSphere Business Integration Event Broker Version 5.0 on the same computer.
- You cannot install more than one instance of either WebSphere MQ Event Broker Version 2.1 or WebSphere Business Integration Event Broker Version 5.0 on a single computer.
- Component names must be unique on the computer, regardless of version.

When you migrate from WebSphere MQ Event Broker Version 2.1 or WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0, you do not have to uninstall the earlier product before you install Version 6.0. You can install WebSphere Event Broker Version 6.0 in a different location on the same computer, migrate your components and resources to Version 6.0, and uninstall WebSphere MQ Event Broker Version 2.1 or WebSphere Business Integration Event Broker Version 5.0 later, when you are sure that you no longer need it.

If you install more than one version on a single computer, and you create multiple components, check that sufficient memory and disk space is available.

The following table shows which Version 2.1 and Version 5.0 components can exist on the same computer as Version 6.0 components:

Can the components exist together on the same computer?	Version 6.0 Message Brokers Toolkit	Version 6.0 Configuration Manager	Version 6.0 broker	Version 6.0 User Name Server
Version 2.1 Control Center	Yes	Yes	Yes	Yes
Version 2.1 Configuration Manager	Yes	Yes	Yes	Yes
Version 2.1 broker	Yes	Yes	Yes	Yes
Version 2.1 User Name Server	Yes	Yes	Yes	Yes[1]
Version 5.0 or Version 5.1 Message Brokers Toolkit	Yes	Yes	Yes	Yes

Can the components exist together on the same computer?	Version 6.0 Message Brokers Toolkit	Version 6.0 Configuration Manager	Version 6.0 broker	Version 6.0 User Name Server
Version 5.0 Configuration Manager	Yes	Yes	Yes	Yes
Version 5.0 broker	Yes	Yes	Yes	Yes
Version 5.0 User Name Server	Yes	Yes	Yes	Yes

Restriction:

1. On distributed systems, only one User Name Server can exist on a single computer.

Coexistence of Version 6.0 components with components from previous versions

With some restrictions, all Version 5.0 components can participate in a Version 6.0 broker domain, and all Version 6.0 components can participate in a Version 5.0 broker domain.

WebSphere MQ Event Broker Version 2.1 components cannot operate with Version 6.0 components at all.

Because Version 6.0 components can coexist with components from Version 5.0, you do not have to migrate all of your components at the same time. You can migrate those components that can coexist with Version 6.0 components in stages that are convenient to you.

The User Name Server has not changed between Version 5.0 and Version 6.0, therefore you can choose not to migrate this component. You can use a Version 5.0 User Name Server with components that you have migrated to Version 6.0.

The following table shows which Version 2.1 and Version 5.0 components can operate with Version 6.0 components, and shows the restrictions that apply.

Can the components operate together in a domain?	Version 6.0 Message Brokers Toolkit	Version 6.0 Configuration Manager	Version 6.0 broker	Version 6.0 User Name Server
Version 2.1 Control Center	No	No	No	No
Version 2.1 Configuration Manager	No	(Not applicable)	No	No
Version 2.1 broker	Yes[1]	Yes[1]	Yes[1]	Yes[1]
Version 2.1 User Name Server	(Not applicable)	No	No	(Not applicable)

Can the components operate together in a domain?	Version 6.0 Message Brokers Toolkit	Version 6.0 Configuration Manager	Version 6.0 broker	Version 6.0 User Name Server
Version 5.0 or Version 5.1 Message Brokers Toolkit	Yes[2]	Yes[3]	Yes	Yes
Version 5.0 Configuration Manager	Yes[3]	(Not applicable)	Yes[4]	Yes
Version 5.0 broker	Yes[5]	Yes	Yes	Yes
Version 5.0 User Name Server	(Not applicable)	Yes	Yes	(Not applicable)

Restrictions:

1. Use of a Version 2.1 broker in a Version 6.0 domain is subject to some restrictions. For full details, see Conditions for a Version 2.1 broker participating in a Version 6.0 broker domain.

2. Sharing resources between Message Brokers Toolkit Version 6.0 and Message Brokers Toolkit Version 5.0 or Version 5.1 is subject to some restrictions. For full details, see "Conditions for using migrated resources with previous versions of the Message Brokers Toolkit"

3. You cannot use a domain user ID to connect a Version 5.0 or Version 5.1 Message Brokers Toolkit to a Version 6.0 Configuration Manager, nor to connect a Version 6.0 Message Brokers Toolkit to a Version 5.0 Configuration Manager.

4. If you deploy through a Version 5.0 Configuration Manager, only message flows and dictionaries are extracted from the broker archive (bar) file.

5. If you want to use a Version 6.0 Message Brokers Toolkit and a Version 5.0 Configuration Manager to deploy to a Version 5.0 broker that is part of a WebSphere Business Integration Event Broker Version 5.0 installation at service level Fix Pack 3 or earlier, apply Fix Pack 5 to the WebSphere Business Integration Event Broker Version 5.0 installation of which the Version 5.0 Configuration Manager is part.

Coexistence with other products

You can create and operate a heterogeneous network of WebSphere MQ Publish/Subscribe brokers and WebSphere Event Broker brokers. You can also migrate WebSphere MQ Publish/Subscribe brokers to WebSphere Event Broker brokers. Follow the instructions in "Migrating publish/subscribe applications" on page 45.

Conditions for using migrated resources with previous versions of the Message Brokers Toolkit

WebSphere Event Broker Version 6.0 can interoperate with components of previous versions. Conditions apply when you use migrated development and deployed resources with previous versions of the Message Brokers Toolkit.

Version 2.1

If you migrate from WebSphere MQ Integrator Broker Version 2.1 (or other supported Version 2.1 products), migrate the following resources:

- Message flows: use the mqsimigratemsgflows command.

For detailed instructions, see "Migrating message flows from Version 2.1" on page 64.

Version 5.0

You do not have to migrate any resources to Version 6.0.

Restrictions after migration

After you have migrated workbench resources to Version 6.0, your use of these resources in the Version 5.0 or Version 5.1 Message Brokers Toolkit is restricted.

Development resources

Resources in Version 5.0, 5.1, and 6.0 formats can coexist in the Version 6.0 workspace.

Resources remain in their original format until you save them in the Version 6.0 Message Brokers Toolkit. After saving, all resources are stored in the Version 6.0 format.

You can no longer use these saved resources in earlier versions of the Message Brokers Toolkit. Therefore you cannot share development resources between the Version 6.0 and previous versions of the Message Brokers Toolkit.

If you expect to continue development of these resources, you must retain a Message Brokers Toolkit at the version that you require.

Deployment resources

You can deploy from the Version 6.0 Message Brokers Toolkit broker archive (bar) files that you created in earlier versions. You can deploy these bar files to brokers that are at previous versions, provided that the development resources referenced in the bar file do not use any features that are available only in Version 6.0.

You can create a new Version 6.0 bar file that includes message flows that were compiled in earlier versions of the workbench.

The Version 6.0 Message Brokers Toolkit can coexist with either the Version 5.0 or Version 5.1 Message Brokers Toolkit on the same computer. However, because you cannot share development resources between the Version 6.0 Message Brokers Toolkit and previous versions of the Message Brokers Toolkit, you might find that it is appropriate to create and maintain a different workspace for the Version 6.0 Message Brokers Toolkit.

The Version 5.0 and Version 5.1 Message Brokers Toolkit cannot coexist on the same computer. Both versions are available only on Windows®.

Migrating from Version 2.1 products

Migrate to WebSphere Event Broker Version 6.0.

You can migrate from the following product:

- WebSphere MQ Event Broker Version 2.1

If you are already using WebSphere MQ Publish/Subscribe, migrate your applications to use the publish/subscribe functions provided by WebSphere Event Broker. Follow the instructions in "Migrating publish/subscribe applications" on page 45.

The instructions apply to all platforms supported by the Version 2.1 products. This section contains the following topics:
- "Planning and pre-migration tasks"
- "Migrating from WebSphere MQ Event Broker Version 2.1 to WebSphere Event Broker Version 6.0" on page 12
- "Post-migration tasks" on page 16

Planning and pre-migration tasks

The following topics describe what to do before migrating from Version 2.1 to Version 6.0:
- "Differences between Version 2.1 and Version 6.0 products"
- "Planning for migration from Version 2.1" on page 10
- "Backing up Version 2.1 resources" on page 11

Differences between Version 2.1 and Version 6.0 products

The following table outlines the main differences between the Version 2.1 and Version 6.0 products:

Version 2.1	Version 6.0	Why has this changed?
Installing the product		
In Version 2.1, there can be only one copy of the product installed on the computer.	You can install the Version 6.0 product alongside the Version 2.1 product and the two products can run side-by-side. The new **mqsimigratecomponents** command allows you to migrate components from Version 2.1 to Version 6.0.	Permitting Version 6.0 to coexist with Version 2.1 allows for easier migration.
Entering commands on Windows		
You enter Version 2.1 commands in a command window.	You enter Version 6.0 commands in the Command Console, which is in the **Start** menu: **Start** → **Programs** → **IBM WebSphere Message Brokers 6.0** → **Command Console**.	Multiple versions of the product can now coexist. The Command Console ensures that the commands that you enter are directed to the correct version.
Configuration Manager Database Requirements		

Version 2.1	Version 6.0	Why has this changed?
You must specify two data sources when creating a Version 2.1 Configuration Manager: a configuration database for message flow source and domain information, and an MRM database for message dictionary source. When deleting a Configuration Manager, there are flags to remove the contents of these repositories (**-n** and **-m** for the configuration and MRM sources respectively).	There are no database requirements on the Configuration Manager. Source data for both message flows and dictionaries is now stored inside the Eclipse workspace. Domain information is stored in an internal configuration repository in the file system on the Configuration Manager computer, and there are commands to back up and restore this repository (**mqsibackupconfigmgr** and **mqsirestoreconfigmgr**). When you delete a Configuration Manager, the **-n** parameter now removes the contents of the internal repository and the **-m** parameter is ignored.	Removing the requirements on an external database simplifies the setup and administration of a Configuration Manager.
Starting the Control Center		
When you start the Control Center, a prompt opens for the connection details of the Configuration Manager.	The Control Center is now called the Message Brokers Toolkit and is based on the Eclipse framework. To connect the Configuration Manager, you must create a connection file by clicking **File** → **New** → **Other** → **Broker Administration** → **Domain**.	Each set of Configuration Manager connection parameters is a file stored within the Eclipse workspace. This allows you to share connection details more easily among multiple users, and allows you to manage multiple domains from a single Message Brokers Toolkit.
Displaying information in the Control Center		
The Version 2.1 Control Center contains a set of tabs that determine the type of information that is shown (for example, assignments, topology, topics, subscriptions).	In Version 6.0, there are two main screens of information (called perspectives): • **Broker Application Development perspective**, which allows you to create and edit message flows. • **Broker Administration perspective**, which shows you the state of the domain, as reported by the Configuration Manager, and allows you to manipulate it. It is possible to switch perspectives using the **Window** menu.	The concept of perspectives is inherited from the Eclipse framework and allows you to switch easily to the information set that is most relevant to what you are doing.

Version 2.1	Version 6.0	Why has this changed?
Modifying message flows		
In Version 2.1, message flows under development are stored in the Configuration Manager. In order to modify them, you must check them out, make your changes and then check them back in.	In Version 6.0, message flows are created and modified in an Eclipse workspace; by default, this is on the file system on which the Message Brokers Toolkit is installed.	Storing development objects in an Eclipse workspace allows you to manage them more effectively. Plug in third-party change-management tools into the Eclipse framework to provide better change control.
Assigning and deploying message flows		
In Version 2.1, message flows were assigned to execution groups. When this is done, the deploy action causes the Configuration Manager to send the assigned message flows to the correct execution group.	In Version 6.0, message flows are packaged in broker archives (bar files). These archives use a zipped file format to envelop a set of deployable objects. To deploy a bar file to an execution group, use the Broker Administration perspective to drag and drop the bar file to the required execution group.	Bar files provide a transferable package of deployment, so that the same message flow logic can be applied to multiple brokers and domains. Similar deployment concepts are used in other products.
Monitoring message flows		
In Version 2.1, traffic lights show the run state of message flows.	In Version 6.0, the Broker Administration perspective has an Alerts view that identifies any broker, execution group or message flow that is not running.	The use of the Alerts view is consistent with other tools in the Eclipse framework and provides an accessible way of notifying you if anything is not processing messages as expected.
User roles		
In Version 2.1, several Windows user groups in the Configuration Manager are used to describe the typical role, and domain access, that each user has (for example, **mqbrkrs**, **mqbrops**, **mqbrtpic**, **mqbrasgn**).	Membership of the **mqbrkrs** group is still required for administration of the broker component, but the **mqsicreateaclentry** command is now used to define specific user or group access control lists (ACLs) for each component (such as brokers and execution groups).	Access control lists (ACLs) in the Configuration Manager allow administrators to control access to the domain at a much finer level of granularity.

Planning for migration from Version 2.1

Before you start:

Before you carry out any migration tasks, back up Version 2.1 resources and read "Coexistence with previous versions and other products" on page 3.

This section explains the different ways in which to migrate from Version 2.1 to Version 6.0. Before carrying out any migration tasks, you should make the following decisions:

1. Decide how you want to migrate the product components:
   a. Find out what is new in Version 6.0. These new and changed functions might affect how you want to use your migrated components in the future.
   b. Decide where you want to migrate the product components. Migrate your components to the same computer or to a second computer. For example, you might want to migrate components to another location to maintain availability during the migration.
   c. Decide when you want to migrate the product components. You might want to preserve some components at the Version 2.1 level of code temporarily, and migrate them later.
   d. Decide the order in which you want to migrate your components. You can migrate components in any order, but your specific circumstances might mean that you need to migrate components in a particular order.

   See "Coexistence with previous versions and other products" on page 3 for information about how Version 6.0 can coexist on the same computer with Version 2.1, and how Version 6.0 components can operate with Version 2.1 components.

2. Decide how you want to use your existing resources with WebSphere Event Broker Version 6.0.

   You migrate your Version 2.1 message flows, and user-defined extensions using the **mqsimigratemsgflows** command. The tooling part of any Version 2.1 user-defined extensions that you have migrated needs to be rewritten in Version 6.0. Decide which user-defined extensions you want to use in Version 6.0 and re-create them in the Version 6.0 tooling.

   When you start using your resources in the Version 6.0 Message Brokers Toolkit, there are restrictions to using these same resources again with the Version 5.0 or Version 5.1 Message Brokers Toolkit. For more information, see "Conditions for using migrated resources with previous versions of the Message Brokers Toolkit" on page 6

3. Decide whether you need to carry out any testing to ensure a successful migration.

   Migrating your development and test domains before migrating your production domain allows you to identify problems and develop a strategy for dealing with further problems.

4. When you are ready to migrate, run the **mqsimigratecomponents** command with the **-c** parameter. This performs a pre-migration check against the Version 2.1 components to ensure that they can be migrated. The pre-migration check identifies potential problems and allows you to correct them before proceeding with migration.

You do not need to change the configuration of a queue manager that is preserved during migration.

Backing up Version 2.1 resources

Before you carry out any migration tasks, back up your Version 2.1 resources:
1. Back up the configuration repository.
2. Back up your broker database tables.
3. Stop any debug sessions and back up your broker application logic (message flow files).

For detailed instructions on how to back up the configuration repository and broker database tables, see your Version 2.1 documentation.

Now that you have backed up your resources, you are ready to perform migration. Go to the appropriate topic for your product:

- Migrating from WebSphere MQ Event Broker

Migrating from WebSphere MQ Event Broker Version 2.1 to WebSphere Event Broker Version 6.0

Before you start

Complete the following task:

- "Preparing to migrate from WebSphere MQ Event Broker Version 2.1" on page 13

and find out more about "Coexistence with previous versions and other products" on page 3.

To migrate a broker domain from WebSphere MQ Event Broker Version 2.1 to WebSphere Event Broker Version 6.0:

1. Close all Version 2.1 Control Center sessions.
2. On each system where one or more brokers are running, perform the following steps for each broker:
   a. Stop the broker by issuing the wmqpsstop command specifying the name of the broker.
   b. Delete the broker by issuing the wmqpsdeletebroker command specifying the name of the broker.
3. Drop the database tables that were used by the brokers that you have just deleted. If a set of broker database tables are in a database by themselves, and you do not intend to use the database after migration, drop the whole database.

   The broker database tables are listed in the *WebSphere MQ Event Broker Version 2.1 Administration Guide*.
4. On the system where the Configuration Manager is running:
   a. Stop the Configuration Manager by issuing the wmqpsstop command for the component ConfigMgr.
   b. Delete the Configuration Manager by issuing the wmqpsdeleteconfigmgr command with the **-n** parameter. This parameter causes the configuration repository to be deleted.
5. On a system where a User Name Server is running:
   a. Stop the User Name Server by issuing the wmqpsstop command for the component UserNameServer.
   b. Delete the User Name Server by issuing the wmqpsdeleteusernameserver command.
6. On each system, uninstall WebSphere MQ Event Broker. Follow the instructions in the appropriate book for your operating system:
   - *WebSphere MQ Event Broker for AIX Version 2.1 Installation Guide*
   - *WebSphere MQ Event Broker for HP-UX Version 2.1 Installation Guide*
   - *WebSphere MQ Event Broker for Solaris Version 2.1 Installation Guide*
   - *WebSphere MQ Event Broker for Windows NT and Windows 2000 Version 2.1 Installation Guide*

   `Windows` On a Windows system, select the option to uninstall including data.

7. On each system, install the components of WebSphere Event Broker Version 6.0 that you require. For detailed instructions on how to perform this task, see the Installation Guide.

8. Re-create your broker domain. As part of this task, perform the following steps, using the information that you recorded previously, where appropriate:

   a. Import the message flows that you want to migrate by issuing the mqsimigratemsgflows command.

   b. Using WebSphere Event Broker commands, re-create and start the Configuration Manager, a User Name Server, and the brokers that you want to migrate.

   c. Start the Message Brokers Toolkit, add each broker to the broker domain and re-create the execution groups that were originally within the broker. Re-create only the execution groups that you want to preserve.

   d. In the workbench, enter and deploy the assignments, topology, and topics configuration data that you want to preserve.

      Deploy migrated message flows to a test environment first. When you are sure that they are working correctly, deploy them to a production environment.

The migration is now complete and the broker domain is ready for use. Delete any queue managers that you no longer need.

The migration does not preserve the subscriptions and retained publications at the brokers. To re-create this information after migration, subscribers must renew their subscriptions and publishers must republish.

Preparing to migrate from WebSphere MQ Event Broker Version 2.1

Complete the preparation tasks before you migrate from Version 2.1.

This topic describes what you must do before you start to migrate a WebSphere MQ Event Broker Version 2.1 broker domain in any of the following tasks:

You must complete the steps in this topic before you perform the task "Migrating from WebSphere MQ Event Broker Version 2.1 to WebSphere Event Broker Version 6.0" on page 12.

1. Make sure that Version 2.1 Control Center users have checked in all WebSphere MQ Event Broker resources.

2. Back up all configuration repository and broker database tables.

3. Decide which message flows you want to migrate and use in the WebSphere Event Broker Version 6.0 broker domain. In a Control Center session, in which all the required message flows are visible in the workspace, export the message flows. Save the exported files in a directory other than the one in which WebSphere MQ Event Broker is installed.

4. Decide which brokers you want to migrate, and which assignments, topology, and topics configuration data you want to preserve.

   For each broker that you want to migrate, and for the associated assignments configuration data that you want to preserve, record the following information (view this information in a Control Center session and record it manually):

   • The name of the broker

   • The name of each execution group within the broker

   • For each execution group within the broker, the name of each message flow that is assigned to the execution group

- For each message flow assigned to an execution group, the following properties:
  - Additional instances
  - Commit count
  - Commit interval
  - Coordinated transaction

For the topology configuration data that you want to preserve, record the following information:

- The name of each collective and the names of the brokers that are in the collective
- All direct connections between brokers

For the topics configuration data that you want to preserve, record the following information:

- The hierarchy of topics, including the name of each topic
- The access control list for each topic

5. Decide which databases and queue managers you want to preserve.

   When you migrate, you will delete all the components of WebSphere MQ Event Broker and drop all the database tables used by the Configuration Manager and the brokers. However, to simplify the task of configuring the new broker domain, preserve the databases and queue managers used by the brokers that you want to migrate.

   If you are migrating a single broker domain from WebSphere MQ Event Broker to WebSphere Event Broker, also preserve the database and queue manager used by the Configuration Manager, and the queue manager used by a User Name Server.

   You do not need to change the configuration of a queue manager that is preserved during migration. You might, however, need to ensure that the WebSphere MQ product code is at the required release and service level to support WebSphere Event Broker. At the same time, ensure that you have installed the other software prerequisites.

6. Decide where you are going to store the development data that is created and maintained in the workbench. Store the data in the local file system, on a shared drive, or in a shared repository that is supported by Eclipse. The instructions for the individual migration tasks assume that you are using the local file system or a shared drive.

7. When you are ready to migrate, run the mqsimigratecomponents command with the **-c** parameter. This performs a pre-migration check against the Version 2.1 components to ensure that they can be migrated. The pre-migration check identifies potential problems, and allows you to correct them before proceeding with migration.

Migrating a message flow

You can migrate the message flows that you have created in WebSphere MQ Event Broker Version 2.1 and use them in WebSphere Event Broker Version 6.0.

You might want to change the message flows that you migrate to take advantage of the new nodes and features that are available in Version 6.0.

You can migrate more than one message flow at one time if you want them to be defined in the same message flow project. You must migrate subflows with the message flows in which they are included to ensure consistent references.

If you have defined more than one message flow with the same name, or the message flow has been exported into more than one export file, the migration task overwrites, without warning, any existing message flow with the next flow that it finds of the same name. Therefore, be careful to avoid conflicts, and to ensure that the most recent version of a message flow that has been defined more than once is the last one that you migrate.

If you have multiple versions of the same message flow, which you use as a subflow in other flows in the same migration directory, the results of the import process are unpredictable.

To migrate a message flow:

1. Before you uninstall Version 2.1, export the message flow or flows from the Control Center using the Version 2.1 tools (see the Version 2.1 documentation for detailed information).

   The migration process is most efficient when all referenced subflows are included in the same export file, therefore export all message flows that you want to migrate to a single message flow project into a single export file.

2. Transfer the export file or files to the new system on which you are running the workbench.

   - Check that the directory in which you store these files does not contain any other files.

   - Store the files that you intend to import into a single message flow project in a separate directory and migrate each directory separately.

   - Make sure that you do not store any files in subdirectories of the project directory, because these files are ignored by the migrate command.

3. If a workbench session is active, close it. You cannot run the migrate command if the workbench is running.

4. At a command prompt, invoke the mqsimigratemsgflows command, specifying the new project name and the directory in which you stored the export files. When the command has completed:

   - The message flows that are contained in the export files in the specified directory are imported into the specified message flow project. If the project already exists, the additional message flows are included with current content, if any. If the project does not exist before you invoke the command, it is created for you. The command is more effective if it creates the message flow project for you.

   - Message flows and subflows are created and their definitions are stored in files named *flow_name*.msgflow.

     After you have imported them, if you want to rename any of these message flows or nodes to conform to your local naming conventions, use only the facilities that are provided by the workbench, in order to preserve consistency and integrity of all references. Do not rename any files within the file system.

5. Check the report file mqsimigratemsgflows.report.txt, which is written to the directory from which you invoked the command. The command provides the following information:

   - The name of each message flow and subflow that has been migrated. If any of these resources had a name that is incompatible with Version 6.0, the command updates the name and all references to that name to ensure consistency. (If you migrate a resource with an invalid name more than once, the correction that is made to the name is always the same.)

- The success or failure of each resource that is migrated.
- An indication of a subflow that cannot be located (its definition is not contained in any of the export files, but it is included in one or more of the migrated message flows). If this problem occurs, find the missing subflow and import it into the appropriate project. If you cannot retrieve the missing subflow for any reason, re-create it with the original name. All affected flows can then link correctly to the new subflow.

   You do not need to repeat the whole export and import process.

6. Start the workbench and switch to the Broker Application Development perspective.
7. Open the message flow project that was created or updated by the migrate command.

   If the project is already open, right-click it and click **Refresh**, then **Rebuild Project** to ensure that the Broker Development view reflects the new content. Rebuild also performs a validation of the message flow project contents.

When you have migrated your resources, see the Version 2.1 post-migration tasks for information about tasks that you might want to perform after migration.

Post-migration tasks

The following topics describe what to do after migrating from Version 2.1 to Version 6.0:
- "Issuing commands from a program after migration"
- "Setting up a command environment" on page 17

Issuing commands from a program after migration

If you have programs that issue WebSphere MQ Event Broker Version 2.1 commands, this topic describes how to run these programs unchanged after you have migrated to WebSphere Event Broker Version 6.0.

UNIX systems:

The names of WebSphere MQ Event Broker Version 2.1 commands start with the prefix "wmqps", not "mqsi". In addition, WebSphere MQ Event Broker commands are stored in a different directory to where WebSphere Event Broker commands are stored. If you have programs, such as shell scripts, that issue WebSphere MQ Event Broker commands, continue to use these programs unchanged after migration by running the shell script migration.sh, which is supplied with WebSphere Event Broker. migration.sh creates the required symbolic links to the WebSphere Event Broker commands.

The following table displays the fully qualified path name of migration.sh for each operating system:

Operating system	Fully qualified path name of migration.sh
AIX®	/usr/opt/mqsi/sample/migration/migration.sh
HP-UX Solaris	/opt/mqsi/sample/migration/migration.sh

migration.sh works by creating symbolic links, so you do not need to run it again after applying a maintenance update to WebSphere Event Broker.

Windows systems:

The names of WebSphere MQ Event Broker Version 2.1 commands start with the prefix "wmqps", not "mqsi". If you have programs, such as batch programs (or batch files), that issue WebSphere MQ Event Broker commands, continue to use these programs unchanged after migration by running the batch program migration.bat, which is supplied with WebSphere Event Broker. migration.bat makes a copy of each WebSphere Event Broker command that has an equivalent WebSphere MQ Event Broker command and gives the copied command its WebSphere MQ Event Broker name.

migration.bat is in the directory *WBIEB*\sample\migration, where *WBIEB* is the fully qualified path name of the directory where you have installed WebSphere Event Broker. You must run migration.bat from this directory.

migration.bat works by copying commands, so you must run it again after every maintenance update you apply to WebSphere Event Broker.

Setting up a command environment

After you have installed the product on any of the distributed systems, you must initialize the environment before you can use any runtime component or command.

Also complete this task if you have migrated to WebSphere Event BrokerVersion 6.0 from a previous version. A profile is provided to help you set up the environment.

Ensure that you use this environment each time you run one of the administrative commands, or start a component; for example, a broker.

For information on setting up your command and runtime environment on a Windows system, see Command environment: Windows platforms

For information on setting up your command and runtime environment on Linux® and UNIX® systems, see Command environment: Linux and UNIX systems

---

Migrating from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0

Migrate product components from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0.

Before you start

Before you start migration:
* Ensure that your installation of WebSphere Business Integration Event Broker Version 5.0 is at service level Fix Pack 4 or later.
* Read about coexistence with previous versions and other products.
* Plan your migration.
* Back up your Version 5.0 resources.

Each of the tasks listed at the end of this topic explains how to migrate a component. Each task also tells you at which point during the migration of that component you must install WebSphere Event Broker Version 6.0.

To migrate a single component, complete the task for that component. To migrate all components, complete all of the tasks listed, but plan your migration so that you apply the tasks as required by your situation. For example, because each task includes steps for installing WebSphere Event Broker Version 6.0, you might want to carry out the tasks in parallel, rather than by completing one task before starting the next task.

The order in which you migrate components is not significant, so complete the tasks in any order. The following list of tasks are presented in a typical order of migration:
- Migrating the Message Brokers Toolkit
- Migrating a Configuration Manager
- Migrating a broker
- Migrating a User Name Server

After you have completed the migration, see the post-migration tasks for information about tasks that you might want to perform after migration.

Planning and pre-migration tasks

Plan your migration and complete pre-migration tasks.

The following topics describe what to do before you migrate from Version 5.0 to Version 6.0:
- "Planning for migration from Version 5.0"
- "Backing up WebSphere Business Integration Event Broker Version 5.0 resources" on page 23

Planning for migration from Version 5.0
Plan the order and extent of the migration of components and resources to Version 6.0.

This section explains how to plan your migration. Take the following steps:
1. Decide how you want to migrate the product components:
    a. Find out what is new in Version 6.0 and learn about new and changed function. These changes might affect how you use your migrated components in the future.
    b. Check the requirements for other products on which Version 6.0 components might have dependencies. Details of prerequisite and other optional products are in both the Installation Guide and the Installation reference section.
    c. Decide where to migrate the product components; to a different location on the same computer or to a second computer. For example, you might want to migrate components to another location to maintain availability during the migration. You might also consider moving the Configuration Manager to another operating system.
       The following topics describe different migration scenarios:
       - "Planning to migrate a small domain from Version 5.0" on page 20
       - "Planning to migrate a large domain from Version 5.0" on page 20

- "Planning to migrate multiple domains from Version 5.0" on page 21
- "Planning to migrate a high-availability domain from Version 5.0" on page 21

d. Decide when to migrate the product components. You might decide to preserve some components at the Version 5.0 level for now, and migrate them later.

e. Decide the order in which to migrate your components. You can migrate components in any order, but your specific circumstances might mean that you need to migrate components in a particular order.

"Coexistence with previous versions and other products" on page 3 shows you how WebSphere Event Broker Version 6.0 can coexist on the same computer with previous versions of the product, and how Version 6.0 components can operate with components from previous versions.

2. Decide how you will use your existing resources with WebSphere Event Broker Version 6.0.

Except for user-defined extensions, you do not need to perform any specific tasks to migrate your development and deployment resources, such as message flow files and broker archive files. You can start using these resources with WebSphere Event Broker Version 6.0 immediately. However, some migration actions are performed automatically when you open or rebuild resources in the toolkit, and some behavioral changes might affect they way in which your message flows and message sets are used; read the guidance provided in "Message flow migration notes" on page 64.

- You must upgrade all Version 5.0 and Version 5.1 user-defined node projects to work with Message Brokers Toolkit Version 6.0. Upgrade a project by cleaning it: click **Project** → **Clean**. When you clean a project, the extension point that is required by Version 6.0 to compile the ESQL files contained in the user-defined extension is created in the project plugin.xml file.

After you start using your current resources in the Message Brokers Toolkit Version 6.0, reuse of those resources is subject to restrictions in Message Brokers Toolkit Version 5.0 or Version 5.1. For more information, see "Conditions for using migrated resources with previous versions of the Message Brokers Toolkit" on page 6.

3. Decide what testing you will do to ensure a successful migration.

The purpose of testing your migration is to identify any problems that might arise during migration. For example, if problems arise you might need to restore some migrated resources to the Version 5.0 level that you backed up before you started the migration, and any post-migration changes to these resources would be lost. Migrating your development and test domains before migrating your production domain allows you to identify problems like this and develop a strategy for dealing with further problems.

4. Optional: When you are ready to migrate, run the mqsimigratecomponents command with the **-c** parameter. This form of the command performs a pre-migration check against the Version 5.0 components to ensure that they can be migrated. The pre-migration check identifies potential problems, which you can correct before proceeding with migration.

5. Optional: Consider adding time to your migration plan to update resources in response to changes in product behavior. Every release includes enhancements and fixes to external behavior, for example a correction to conform to an external standard. If your resources depend on undocumented or incorrect behavior, you might need to make changes and test these resources to understand the implications in your business scenarios.

After you have planned your migration, back up your resources.

Planning to migrate a small domain from Version 5.0:

Migrate a small domain that extends over a small number of computers.

Typically, a small domain has many or all components installed on a single computer. In this situation, the simplest approach is to migrate all the components on that single computer at the same time. If you cannot migrate in one step, install WebSphere Event Broker Version 6.0 at a different location on the same computer, or on a second computer, and migrate each component separately. You can stage the migration because Version 5.0 components can coexist with Version 6.0 components within a single domain. See "Coexistence with previous versions and other products" on page 3 for more information.

Migrate components in the order that best suits your environment. For example, a typical order in which to migrate the three major components might be as follows:
1. Instances of the Message Brokers Toolkit
2. Configuration Manager
3. Brokers

If you migrate instances of the Message Brokers Toolkit first, you can begin to develop message flows that use features that are new in Version 6.0. If you migrate the Configuration Manager, administrators can prepare your Version 6.0 broker domain for deployment. Finally, when you migrate the brokers, you can run your new message flows.

Depending on the goals of migration, the order in which you migrate components can be different. For example, if the primary reason for migration is to make use of new broker functions, you might want to migrate the brokers first. Or if you plan to move your Configuration Manager to a new operating system, you might want to migrate the Configuration Manager first to discover any operating system or environment issues early on.

Before you start to perform migration, back up your resources.

Planning to migrate a large domain from Version 5.0:

Migrate a large domain that contains many components and extends over many computers.

On a large domain, you might not be able to migrate all components at the same time. In this situation, you can split the migration of the components, for example:
1. Migrate a test Message Brokers Toolkit and a test Configuration Manager
2. Migrate a test broker
3. Migrate more instances of the Message Brokers Toolkit
4. Migrate more brokers
5. Migrate the remaining components

The migration order is flexible because Version 6.0 components can coexist with Version 5.0 components within a single domain. However, coexistence does enforce some restrictions. In particular, these restrictions will affect you if you plan to stagger the migration of several instances of the Message Brokers Toolkit, because some files cannot be read using Message Brokers Toolkit Version 5.0 or Version 5.1

after they have been saved using Message Brokers Toolkit Version 6.0. See "Coexistence with previous versions and other products" on page 3 for more information.

Before you start to perform migration, back up your resources.

Planning to migrate multiple domains from Version 5.0:

Migrate multiple domains that perform different functions in your environment.

A typical configuration for WebSphere Event Broker is to use a set of three or more domains, consisting of development, test, and production domains.

Development domain
> In a development domain, message flow developers create message flows and unit test them in a sandbox environment. The brokers in the domain are not responsible for handling business-critical data.

Test domain
> Message flows in the development domain are eventually promoted to a test domain, where message flows are tested against recent, but not live, production data in a realistic broker configuration.

Production domain
> When message flows in the test domain are deemed to be sufficiently robust, they are promoted to the production domain. This domain is responsible for performing actual business transactions, and message flows in the domain work with live data. Non-critical updates to production flows usually take place only at predefined service intervals.

Development and test domains must be migrated before production domains. Migrating the development domain first minimizes potential downtime associated with any migration activity. In addition, development domains typically require access to new broker functions before test and production domains. When you develop message flows that make use of new functions, you must migrate the test and production domains before you deploy the new message flows to them.

Because each Message Brokers Toolkit can administer multiple domains, be careful when migrating instances of the Message Brokers Toolkit not to affect any domains that have not been migrated yet.

Message Brokers ToolkitVersion 6.0 can administer Version 5.0 domains; similarly, Message Brokers ToolkitVersion 5.0 can administer Version 6.0 domains

Before you start to perform migration, back up your resources.

Planning to migrate a high-availability domain from Version 5.0:

Special considerations apply when you are migrating a high-availability domain.

High availability is the requirement for a system to be running all, or for a very high proportion of, the time. Providing a high-availability environment involves having multiple brokers so that some brokers can be used for backup purposes. The entire production domain might be duplicated so that the backup system can be switched on in the event of a problem.

When you migrate, a high-availability environment presents a potential problem because a vulnerable period exists between stopping the previous version of the

product, and installing and starting WebSphere Event Broker Version 6.0. One solution is to install and configure WebSphere Event Broker Version 6.0 while the previous version of the product is running in parallel, and then switch to WebSphere Event Broker Version 6.0. Install WebSphere Event Broker Version 6.0 on the same computers that are running the previous version of the product, or on a different set of computers.

The following sections of this topic provide a general framework for how you might want to migrate your high-availability domain. Adjust this guidance to suit your specific circumstances and refer to the product-specific migration topics for detailed migration instructions.

Preparing to migrate a high-availability domain:
To achieve high-availability migration, your environment must be configured correctly. Complete the following tasks (production and test domains are described in "Planning to migrate multiple domains from Version 5.0" on page 21):

1. Ensure that the production domain that you want to migrate is configured for high availability. You must have at least three brokers supporting the executing applications, so that when broker 1 is being migrated, brokers 2 and 3 can provide backup support for each other. This setup is typical for a high-availability domain at all times, and is not specific to migration.

2. Ensure that the test domain is identical to the production domain. If your domains are set up in this way, you can identify any problems during the migration of the test domain and resolve them before you migrate the production domain.

Preparing to migrate the test domain:

By migrating the test domain first, you can ensure that operation is successful after migration, and solve any problems before migrating the production domain. Test the migration of your resources to WebSphere Event Broker Version 6.0 for each functional area in turn. The following steps describe one way in which to migrate the test domain to minimize disruption. Refer to the product-specific migration topics for detailed instructions.

1. After you have backed up your resources and installed Version 6.0, create a Version 6.0 Configuration Manager and broker on as many computers as are required to mirror the production domain.

2. Move one functional area to Version 6.0 by migrating the Message Brokers Toolkit and the resources that are used within that area. Test that the migrated resources work as expected by deploying them and checking the results. Resolve any problems, then migrate the remaining functional areas. If you migrate the functional areas one by one, existing development can continue on the previous version of the product.

3. In the same way that you migrate the functional areas individually, migrate the brokers individually to Version 6.0. When you have migrated one broker, test that your resources still work as expected by deploying the migrated resources for which this broker is responsible. Resolve any problems and repeat for the remaining brokers. Migrating the brokers one at a time means that only small changes are made at each stage. If problems occur, other brokers that have not been migrated yet can process requests while the migration problems are resolved.

Preparing to migrate the production domain:
After you have migrated the test domain and resolved any problems, prepare to migrate the production domain. Before you migrate your resources, configure an

administration system with Message Brokers Toolkit Version 6.0. You can then administer the new Version 6.0 components. Refer to the product-specific migration topics for detailed instructions.

When you are ready to perform migration, back up your resources.

Backing up WebSphere Business Integration Event Broker Version 5.0 resources

Back up your resources before migration.

Before you start:

Read the following planning topics:

- "Planning to migrate a small domain from Version 5.0" on page 20
- "Planning to migrate a large domain from Version 5.0" on page 20
- "Planning to migrate multiple domains from Version 5.0" on page 21
- "Planning to migrate a high-availability domain from Version 5.0" on page 21

Before you carry out any migration tasks, back up your WebSphere Business Integration Event Broker Version 5.0 resources:

1. Back up the configuration repository database. Use the documented procedures that are provided by DB2.
2. Back up the broker database tables. Use the documented procedures that are provided by your database supplier.
3. Back up your Message Brokers Toolkit workspace and resources; for example, message flow files and broker archive (bar) files.
   - If you manage your workspace resources in a shared repository, for example CVS, follow standard backup procedures for safeguarding versions.
   - If you maintain your workspace resources on a local or shared disk, copy your workspace directory to a different location.

For detailed instructions on how to back up these resources, see the WebSphere Business Integration Event Broker Version 5.0 information center.

After you have backed up your WebSphere Business Integration Event Broker Version 5.0 resources, proceed to one of the following topics:

- "Migrating from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0" on page 17

Migrating the Message Brokers Toolkit from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0

Migrate the Message Brokers Toolkit Version 5.0.

If you are using Message Brokers Toolkit Version 5.1, follow the instructions in this topic, replacing all references to "Message Brokers Toolkit Version 5.0" with "Message Brokers Toolkit Version 5.1".

When you start using your resources in Message Brokers Toolkit Version 6.0, restrictions apply to the use of these same resources again with Message Brokers

Toolkit Version 5.0 or Version 5.1. For more information, see "Conditions for using migrated resources with previous versions of the Message Brokers Toolkit" on page 6.

To migrate Message Brokers Toolkit Version 5.0 to Version 6.0:

1. Install Version 6.0 in a different location from WebSphere Business Integration Event Broker Version 5.0.

2. When you start the Message Brokers Toolkit Version 6.0 for the first time, you are prompted to enter a workspace location. Enter the directory where the Message Brokers Toolkit Version 5.0 workspace that you want to migrate is located. In Version 5.0, the default location is *install_dir*/eclipse/workspace.

   - Version 5.0 message flows are converted to the Version 6.0 format automatically when you save them for the first time. Message Brokers Toolkit Version 5.0 cannot read message flows that have been saved in Version 6.0 format.

   - If your message flows use WebSphere MQ Everyplace® (MQe) nodes, follow the instructions in "Migrating a message flow that contains WebSphere MQ Everyplace nodes" on page 67.

3. Clean and rebuild your workspace.

4. To migrate user-defined nodes from Version 5.0, import the user-defined node project into the Version 6.0 workbench and rebuild the project. If you are migrating user-defined nodes from Version 5.0, perform the following additional steps:

   a. Modify the `<requires>` element in the plugin.xml file in the user-defined node project root to match the following example. Ensure that you delete all extraneous entries from the Version 5.0 plugin.xml file so that it matches the following example exactly:

   ```
   <requires>
    <import match="greaterOrEqual" plugin="com.ibm.etools.mft.api" version="6.0.0"/>
   </requires>
   ```

   b. Modify the "org.eclipse.help.contexts" extension in the same plugin.xml file to match the following example:

   ```
   <extension point="org.eclipse.help.contexts">
    <contexts file="HelpContexts.xml"/>
   </extension>
   ```

   c. Restart the Message Brokers Toolkit using the -clean option.

If you have configured MCA users on the WebSphere MQ channels to the Configuration Manager, you might encounter problems when you try to connect to the migrated Configuration Manager from Message Brokers Toolkit Version 6.0. To resolve these problems, configure access control lists for users who are running the Message Brokers Toolkit Version 6.0. Follow the instructions in Considering security for the workbench.

When you have migrated the Message Brokers Toolkit, migrate the rest of the components:

- Migrate the Configuration Manager
- Migrate the broker
- Migrate the User Name Server

When you have completed these tasks, see "Post-migration tasks" on page 38 for information about tasks that you might want to perform after migration.

Migrating a Configuration Manager from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0

Follow the instructions to migrate the Configuration Manager Version 5.0.

In WebSphere Event Broker Version 6.0, the Configuration Manager no longer stores the domain configuration in an external database; it uses an internal repository instead. The Configuration Manager is also available on more operating systems than in WebSphere Business Integration Event Broker Version 5.0.

From the following list of topics in this section, select the topic that is appropriate to your environment:
- Migrating on the same computer
- Migrating to a different computer that has DB2 installed
- Migrating to a different computer that does not have DB2 installed

Migrating a Version 5.0 Configuration Manager to Version 6.0 on the same computer

Migrate a Version 5.0 Configuration Manager to Version 6.0 on the same system.

Before you start

Read about "Coexistence with previous versions and other products" on page 3.

If the Configuration Manager runs in a locale that is not listed in the Locales topic, check that the code page is one of the supported code pages and that the locale is set up correctly.

To migrate a Version 5.0 Configuration Manager to Version 6.0:
1. Install WebSphere Event Broker Version 6.0 in a location other than where Version 5.0 is installed.
2. Stop the Version 5.0 Configuration Manager.
3. Use the mqsimigratecomponents command to migrate your Version 5.0 Configuration Manager.
4. Start the Version 6.0 Configuration Manager using the **mqsistart** command.

If you do not have additional hardware available, but you want to confirm the successful migration of the Configuration Manager before you remove the Version 5.0 Configuration Manager, migrate the Configuration Manager to a different location on the same computer. If you use this method, you can also change the name of the Configuration Manager.

Complete the following steps:
1. Install WebSphere Event Broker Version 6.0 in a location other than where Version 5.0 is installed.
2. Stop the Version 5.0 Configuration Manager.
3. Create a Version 6.0 Configuration Manager using the mqsicreateconfigmgr command. Make sure that the Version 6.0 Configuration Manager has a different name and a different queue manager name to the Version 5.0 Configuration Manager. Use the **-n** parameter of the mqsicreateconfigmgr command to specify the Version 5.0 Configuration Manager database.

4. Configure WebSphere MQ to allow the Version 6.0 Configuration Manager to communicate with the broker network. For example, you might need to configure channels and transmission queue definitions.

5. Start the Version 6.0 Configuration Manager using the **mqsistart** command.

6. Deploy the complete topology using either the Message Brokers Toolkit or the mqsideploy command. This action subscribes all brokers in the domain with the new Version 6.0 Configuration Manager, so that the brokers publish their status messages to the correct queue manager. When deployment is complete, the brokers are managed by the Version 6.0 Configuration Manager on the new system; do not use the previous Configuration Manager.

If you have configured MCA users on the WebSphere MQ channels to the Configuration Manager, you might encounter problems when you try to connect to the migrated Configuration Manager from the Message Brokers Toolkit Version 6.0. To resolve these problems, configure access control lists for users who are running the Message Brokers Toolkit Version 6.0. Follow the instructions in Considering security for the workbench.

Before you migrate your brokers, or make any configuration changes, ensure that your Configuration Manager has been migrated correctly by performing the following checks:

1. Start the migrated Configuration Manager.

2. Start the Message Brokers Toolkit and connect it to the Configuration Manager's domain.

3. Check the list of domain components and see if any errors are produced.

4. Check the list of alerts for the domain and see if anything has stopped.

5. Check that you can open the Event Log.

6. Optional: If you have a broker running, stop and start a message flow, and check for a successful response; check that the Alerts view is updated when the message flow is stopped, and clear when the message flow is started.

If these steps are successful, you have successfully migrated your Configuration Manager. If they are unsuccessful, resolve any problems before you migrate further components or make configuration changes.

When you have migrated the Configuration Manager, migrate the rest of the components:

- Migrate the Message Brokers Toolkit
- Migrate the broker
- Migrate the User Name Server

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Migrating a Version 5.0 Configuration Manager to Version 6.0 on a different computer that has DB2 installed

Migrate a Version 5.0 Configuration Manager to Version 6.0 on another system.

Before you start

Read about "Coexistence with previous versions and other products" on page 3.

If the Configuration Manager runs in a locale that is not listed in the Locales topic, check that the code page is one of the supported code pages and that the locale is set up correctly.

An existing Windows Configuration Manager can be migrated to a Version 6.0 Configuration Manager on any of the supported operating systems through a JDBC Type 4 Universal DB2® connection.

To migrate a Configuration Manager to a different computer that has a JDBC Type 4 Universal DB2 connection installed, complete the following steps:

1. Install WebSphere Event Broker Version 6.0 on the computer to which you are migrating your Version 5.0 Configuration Manager.

2. Modify the environment in which you are going to create the Version 6.0 Configuration Manager. On z/OS®, update the BIPCPROF file and submit the BIPGEN job when you have completed these updates. On all other operating systems, update the local environment.

   a. Add the following files to the CLASSPATH:

      db2_install/jcc/classes/sqlj.zip
      db2_install/jcc/classes/db2jcc.jar
      db2_install/jcc/classes/db2jcc_javax.jar
      db2_install/jcc/classes/db2jcc_license_cisuz.jar

   b. Add the following entry to the environment variable for libraries (for example, LIBPATH on z/OS):

      db2_install/jcc/lib

   c. Add the following content to the PATH:

      db2_install/jcc/bin

   where *db2_install* is the directory in which DB2 is installed at your location (for example /usr/lpp/db2710/db2710).

3. Stop the Version 5.0 Configuration Manager.

4. Create a Version 6.0 Configuration Manager on the second computer by issuing the mqsicreateconfigmgr command or run the BIPCRCM job on z/OS. Specify the database name, user name, and password that are required to access the Version 5.0 Configuration Manager database on Windows. For example:

   -u (userid)
   > The user ID for the Configuration Manager configuration repository database.

   -p (password)
   > The password associated with the database user ID.

   -n (database name)
   > The identification of the database, in the following format:
   > `//server:port/database_name`
   >
   > where:
   > - *server* is the IP address of the computer where the database exists.
   > - *port* is the port number of DB2 on that computer.
   > - *database_name* is the name of the database (for example MQSICMDB)
   >
   > For example: `//9.20.235.197:50000/MQSICMDB`.
   >
   > To determine the port number:
   > a. Open the DB2 Control Center and right-click the DB2 instance.

b. Select **Setup Communications...**.

c. Enter the user ID and password, if required.

The properties box contains the address and port number for this DB2 instance.

To maintain uniqueness in the WebSphere MQ network, use different queue manager names for the two Configuration Manager components.

When you create the Version 6.0 Configuration Manager, domain information from the Version 5.0 Configuration Manager database is copied to the Version 6.0 Configuration Manager internal repository, therefore it might take a few minutes to migrate the database.

5. On the second computer, configure WebSphere MQ to allow the Version 6.0 Configuration Manager to communicate with the broker network. For example, you might need to configure channels, transmission queues, and remote queue manager definitions.

6. Start the Version 6.0 Configuration Manager.

7. Deploy the complete topology using either the Message Brokers Toolkit or the mqsideploy command. This action subscribes all brokers in the domain with the new Version 6.0 Configuration Manager, so that the brokers publish their status messages to the correct queue manager. When deployment is complete, the brokers are managed by the Version 6.0 Configuration Manager on the new system; do not use the previous Configuration Manager.

Before you migrate your brokers, or make any configuration changes, perform the following checks to ensure that your Configuration Manager has been migrated correctly:

1. Start the migrated Configuration Manager.

2. Start the Message Brokers Toolkit and connect it to the domain of the migrated Configuration Manager.

3. Check the list of domain components and see if any errors are produced.

4. Check the list of alerts for the domain and see if anything has stopped.

5. Check that you can open the Event Log.

6. Optional: If you have a broker running, stop and start a message flow, and check for a successful response; check that the Alerts view is updated when the message flow is stopped, and clear when the message flow is started.

If these steps are successful, you have successfully migrated your Configuration Manager. If they are unsuccessful, resolve any problems before you migrate further components or make configuration changes.

When you have migrated the Configuration Manager, migrate the rest of the components:

- Migrate the Message Brokers Toolkit
- Migrate the broker
- Migrate the User Name Server

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Migrating a Version 5.0 Configuration Manager to Version 6.0 on a different computer that does not have DB2 installed

Migrate a Version 5.0 Configuration Manager to Version 6.0 on another system.

Before you start

Read about "Coexistence with previous versions and other products" on page 3.

If the Configuration Manager runs in a locale that is not listed in the Locales topic, check that the code page is one of the supported code pages and that the locale is set up correctly.

To migrate a Configuration Manager to a different computer that does not have DB2 or a JDBC client for DB2 installed, complete the following steps:

1. Migrate the Version 5.0 Configuration Manager to Version 6.0 on the same computer:
   a. Install WebSphere Event Broker Version 6.0 in a location other than where WebSphere Business Integration Event Broker Version 5.0 is installed.
   b. Stop the Version 5.0 Configuration Manager.
   c. Launch a Version 6.0 Command Console and enter the mqsimigratecomponents command to migrate your Version 5.0 Configuration Manager.
   d. Start the Version 6.0 Configuration Manager.

      When you start the Version 6.0 Configuration Manager for the first time, it automatically detects domain information in the DB2 database of the Version 5.0 Configuration Manager, and migrates it into the internal repository of the Version 6.0 Configuration Manager. You do not need to do anything, and the DB2 database is not modified.
2. Stop the Version 6.0 Configuration Manager, and make a copy of its internal repository using the mqsibackupconfigmgr command.
3. Install WebSphere Event Broker Version 6.0 on the second computer.
4. On the second computer, create a Version 6.0 Configuration Manager by either issuing the mqsicreateconfigmgr command or running the BIPCRCM job on z/OS.

   You must maintain unique queue manager names in your WebSphere MQ network. When you create the Version 6.0 Configuration Manager, choose a different queue manager name.
5. On the second computer, configure WebSphere MQ to allow the Version 6.0 Configuration Manager to communicate with the broker network. For example, you might need to configure channels, transmission queues, and remote queue manager definitions.
6. On the second computer, issue the mqsirestoreconfigmgr command (or the BIPRSCM job on z/OS) to overwrite the contents of the empty Version 6.0 Configuration Manager repository with the repository that you backed up from the original computer.
7. On the second computer, start the Version 6.0 Configuration Manager.
8. Deploy the complete topology using either the Message Brokers Toolkit or the mqsideploy command. This action subscribes all brokers in the domain with the new Version 6.0 Configuration Manager, so that the brokers publish their status messages to the correct queue manager. When deployment is complete, the brokers are managed by the Version 6.0 Configuration Manager on the new system; do not use the previous Configuration Manager.

Before you migrate your brokers, or make any configuration changes, ensure that your Configuration Manager has been migrated correctly by performing the following checks:

1. Start the migrated Configuration Manager.
2. Start the Message Brokers Toolkit and connect it to the Configuration Manager domain.
3. Check the list of domain components and see if any errors are produced.
4. Check the list of alerts for the domain and see if anything has stopped.
5. Check that you can open the Event Log.
6. Optional: If you have a broker running, stop and start a message flow, and check for a successful response; check that the Alerts view is updated when the message flow is stopped, and clear when the message flow is started.

If these steps are successful, you have successfully migrated your Configuration Manager . If they are unsuccessful, resolve any problems before you migrate further components or make configuration changes.

When you have migrated the Configuration Manager, migrate the rest of the components:

- Migrate the Message Brokers Toolkit
- Migrate the broker
- Migrate the User Name Server

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Migrating a broker from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0

Follow the instructions to migrate the broker.

To migrate a broker from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0, see the appropriate topic for your operating system:

- Distributed systems
- z/OS

Migrating a Version 5.0 broker to Version 6.0 on distributed operating systems

Migrate a broker to use the enhanced facilities available in a Version 6.0 broker domain.

If you stop the broker, you can migrate it immediately to the new version on the same computer. If you prefer not to stop the broker to avoid problems for your business applications, or if you want to reproduce the broker capability on another computer, you can move the Version 5.0 broker's application logic onto a separate Version 6.0 broker.

If you are using the Version 5.1 Message Brokers Toolkit, replace all references to "Version 5.0" with "Version 5.1".

Select the topic that is appropriate to your environment.

- "Migrating a Version 5.0 broker to Version 6.0"
- "Migrating a Version 5.0 broker's application logic to another installation"

Migrating a Version 5.0 broker to Version 6.0:
To migrate a Version 5.0 broker on distributed operating systems to Version 6.0 on the same computer, complete the following steps:

1. Install WebSphere Event Broker Version 6.0 in a different location from WebSphere Business Integration Event Broker Version 5.0.
2. Stop the Version 5.0 broker.
3. Stop any channels to the Version 5.0 broker.
4. Update the ODBC definitions file to Version 6.0 format by following the instructions in "Updating your ODBC definitions when migrating" on page 32.
5. Set up the correct Version 6.0 command environment.

   - `Linux` `UNIX` On Linux and UNIX systems, run the environment profile mqsiprofile.

   - `Windows` On Windows, open a Command Console from the **Start** menu.
6. Enter the mqsimigratecomponents command to migrate the broker. For example:

   ```
   mqsimigratecomponents Broker1
   ```
7. If you are running Version 6.0 on a 64-bit operating system, make the following changes to your environment:
   - You no longer have to export 32-bit libraries in the LIBPATH environment variable. Remove all variable settings similar to the following settings:

     ```
     export LD_LIBRARY_PATH=[DB2instanceDir]/sqllib/lib32:$LD_LIBRARY_PATH
     export SHLIB_PATH=$ORACLE_HOME/lib32:$SHLIB_PATH
     ```
   - If you want to continue to use 32-bit execution groups, and you use DB2 for your broker database, set the appropriate environment variable for the DB2 version that you have installed.

     For DB2 Version 8:

     ```
     MQSI_LIBPATH32=$MQSI_LIBPATH32:[DB2InstanceDir]/sqllib/lib
     ```

     For DB2 Version 9:

     ```
     MQSI_LIBPATH32=$MQSI_LIBPATH32:[DB2InstanceDir]/sqllib/lib32
     ```

     If your broker will support only 64-bit execution groups, this variable is not required.
8. Start the Version 6.0 broker using the mqsistart command.

Migrating a Version 5.0 broker's application logic to another installation:

Before you start

To complete this task you must have either a Version 5.0 or a Version 6.0 Message Brokers Toolkit available.

Before migrating a broker, ensure that you do not have any aggregations in progress. When you migrate a broker to Version 6.0, any live data that is being stored for aggregations in progress will be lost.

If the broker runs in a locale that is not listed in the Locales topic, check that the code page is one of the supported code pages and that the locale is set up correctly.

To migrate a Version 5.0 broker on distributed operating systems to Version 6.0 at a different location on the same computer, or to a different computer, complete the following steps:

1. Install WebSphere Event Broker Version 6.0 either as a new instance on the computer where WebSphere Business Integration Event Broker Version 5.0 is installed, or on a different computer. For detailed instructions, see the Installation Guide.
2. Update the ODBC definitions file to Version 6.0 format by following the instructions in "Updating your ODBC definitions when migrating."
3. Create a Version 6.0 broker; give it a name that is different from the name of the Version 5.0 broker.

   Brokers cannot share queue managers. You cannot migrate a broker to a queue manager that is already being used by an existing broker.
4. Start the Version 6.0 broker using the **mqsistart** command.
5. Add the new broker to the domain by completing the following tasks:
   a. Create a Configuration Manager
   b. Create a domain connection
   c. Connect to the broker domain
   d. Add a broker to a broker domain
6. Write a list of the execution groups that you have on the Version 5.0 broker, and create these same execution groups on the new broker. Use either the Version 5.0 or Version 6.0 Message Brokers Toolkit to complete this step.
7. Deploy the message flows of the Version 5.0 broker to the Version 6.0 broker. Use either the Version 5.0 or Version 6.0 Message Brokers Toolkit to complete this step.
8. Configure any other relevant properties of the Version 5.0 broker on the Version 6.0 broker. For example, you might need to configure properties that you set using the mqsichangeproperties command, or as a result of using certain nodes, such as Publication or SCADA nodes.
9. Stop the Version 5.0 broker.
10. Remove the Version 5.0 broker from the workbench.
11. Redeploy the topology.
12. Delete the Version 5.0 broker.

When you have migrated the broker, migrate the rest of the components:
- Migrate the Message Brokers Toolkit
- Migrate the Configuration Manager
- Migrate the User Name Server

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Updating your ODBC definitions when migrating:

As part of migrating a broker, you create ODBC definitions for broker databases that specify appropriate database drivers for the new product version.

Complete this update before you run the mqsimigratecomponents command.

Follow the instructions provided for your operating system:

Windows systems

The operating system stores ODBC definitions in a single repository of ODBC data source names. Change Oracle and Sybase data source definitions according to the instructions in "Changing the 32-bit ODBC connection and XA resource manager definitions for a broker migrated from Version 2.1 or Version 5.0 to Version 6.0"

You do not need to change DB2, Informix, and SQL Server data source definitions.

Linux and UNIX systems

ODBC definitions are stored in a text file. For previous versions of the product, the file .odbc.ini is located in /var/mqsi/odbc/. When working with Version 6.0, create a new text file in this directory, but do not replace the existing file to avoid affecting existing brokers. Set the environment variable ODBCINI to point to the new file; perform this step as part of the setup for a Version 6.0 command environment.

In your new Version 6.0 ODBC configuration file, ensure that you list all of the data sources that the brokers that you want to migrate will use. For information on how to create a new ODBC configuration file, see Connecting to a database from Linux and UNIX systems.

If you are using XA to enable coordinated transactions, you must also update your qm.ini file for WebSphere MQ, as described in "Changing the 32-bit ODBC connection and XA resource manager definitions for a broker migrated from Version 2.1 or Version 5.0 to Version 6.0"

If you revert to a previous version of WebSphere Event Broker, you must reverse the changes that you make to the ODBC definitions. Update the odbc.ini after you have run the mqsimigratecomponents command, but before you restart the broker. If you are reverting to Version 5.0, follow the instructions in "Restoring components and resources to Version 5.0" on page 40.

Changing the 32-bit ODBC connection and XA resource manager definitions for a broker migrated from Version 2.1 or Version 5.0 to Version 6.0:

Update definitions for Oracle and Sybase databases.

WebSphere Event Broker Version 6.0 includes later versions of the ODBC drivers that are supplied by DataDirect Technology (formerly Merant). These ODBC drivers are for Oracle and Sybase databases. A new version of the DataDirect Driver Manager for Linux and UNIX systems is also included. You must make a number of changes as a result of the later versions of the ODBC drivers for Oracle and Sybase databases.

Make the required changes before you run the Version 6.0 mqsimigratecomponents command. If you are restoring to an earlier version, keep the new definitions until after you have finished running the mqsimigratecomponents command. The following sections describe the changes in detail.

- `Linux` `UNIX` If you are migrating a broker from Version 2.1 or Version 5.0 to Version 6.0 on Linux or UNIX systems, create a new ODBC configuration file that contains definitions for each of your existing data sources as described in "Changing the 32-bit ODBC connection definitions on Linux and UNIX systems" on page 34.

  You must also change the XA resource manager definition for the database if you have a message flow that is running in the broker and updates an Oracle or Sybase database within a global unit of work that is coordinated by the broker

queue manager. Follow the instructions provided in "Changing the XA resource manager definitions on Linux and UNIX systems."

- Windows If you are migrating a broker from Version 2.1 or Version 5.0 to Version 6.0 on a Windows system, update the ODBC definition for each broker and the user data source name as described in "Changing the 32-bit ODBC connection definitions on Windows systems" on page 35. Complete this task for Oracle and Sybase databases only.

  You must also change the XA resource manager definition for the database if you have a message flow that is running in the broker and updates an Oracle or Sybase database within a global unit of work that is coordinated by the broker queue manager. Follow the instructions provided in "Changing the XA resource manager definitions on Windows systems" on page 35.

- If you are restoring a broker from Version 6.0 back to Version 2.1 or Version 5.0, you must reverse these changes.

Changing the 32-bit ODBC connection definitions on Linux and UNIX systems:

Linux UNIX Create a new ODBC definitions file by following the instructions in Connecting to a database from Linux and UNIX systems. Do not change the existing definition file.

Before you run Version 6.0 commands, check that your ODBCINI environment variable points to the new definitions file and not to the existing file.

Changing the XA resource manager definitions on Linux and UNIX systems:

Linux UNIX To change the XA resource manager definitions on a Linux or UNIX system, edit the queue manager configuration file (qm.ini) of the broker queue manager. This file is in the queue manager's directory.

In the XAResourceManager stanza for each Oracle and Sybase database that participates in a global unit of work, coordinated by the broker queue manager, change the entry for the switch file. The following tables specify what you need to change for each broker operating system and database management system (DBMS).

AIX Complete the following changes on AIX.

DBMS	Change ...	To ...

Oracle	`SwitchFile=`
	`  /usr/opt/mqsi/merant/lib/UKor8dtc16.so`
	
	or
	
	`SwitchFile=`
	`  /usr/opt/mqsi/merant/lib/UKor8dtc18.so`
		`SwitchFile=`
		`  `*`install_dir`*`/merant/lib/UKor8dtc20.so`

Sybase	`SwitchFile=`
	`  /usr/opt/mqsi/merant/lib/UKase16.so`
	
	or
	
	`SwitchFile=`
	`  /usr/opt/mqsi/merant/lib/UKase18.so`
		`SwitchFile=`
		`  `*`install_dir`*`/merant/lib/UKase20.so`

Solaris Linux Complete the following changes on Solaris and Linux.

DBMS	Change ...	To ...

Oracle	`SwitchFile=`
	`  /opt/mqsi/merant/lib/UKor8dtc16.so`
	
	or
	
	`SwitchFile=`
	`  /opt/mqsi/merant/lib/UKor8dtc18.so`
		`SwitchFile=`
		`  install_dir/merant/lib/UKor8dtc20.so`

Sybase	`SwitchFile=`
	`  /opt/mqsi/merant/lib/UKase16.so`
	
	or
	
	`SwitchFile=`
	`  /opt/mqsi/merant/lib/UKase18.so`
		`SwitchFile=`
		`  install_dir/merant/lib/UKase20.so`

HP-UX Complete the following changes on HP-UX.

DBMS	Change ...	To ...
Oracle	`SwitchFile=` `  /opt/mqsi/merant/lib/libSwitchOracle.sl`	`SwitchFile=` `  install_dir/merant/lib/UKor8dtc20.sl`
Sybase	`SwitchFile=` `  /opt/mqsi/merant/lib/libSwitchSybase.sl`	`SwitchFile=` `  install_dir/merant/lib/UKasedtc20.sl`

Changing the 32-bit ODBC connection definitions on Windows systems:

Windows To change the ODBC connection definitions on a Windows system:
1. Open the ODBC Data Source Administrator window.
2. Open the System DSN page.
3. For each Oracle and Sybase database that is accessed by the broker, associate the data source name with the new ODBC driver:
   a. Delete the data source by clicking **Remove**.
   b. Re-create the data source with the new ODBC driver by clicking **Add**.

The following table displays the name of the new ODBC driver for each database management system (DBMS).

DBMS	New ODBC driver
Oracle	MQSeries® DataDirect Technologies 5.0 32-BIT Oracle
Sybase	MQSeries DataDirect Technologies 5.0 32-BIT Sybase Wire Protocol

Changing the XA resource manager definitions on Windows systems:

Windows To change the XA resource manager definitions on a Windows system:
1. Open the **Properties** window of the broker queue manager using the WebSphere MQ Services snap-in.
2. Open the **Resources** page.
3. For each Oracle and Sybase database that participates in a global unit of work, coordinated by the broker queue manager, change the contents of the **SwitchFile** field. The following table specifies what you need to change for each database management system (DBMS).

WBIMB represents the fully qualified path name of the directory in which you have installed WebSphere Event Broker, and WMQIB represents the fully qualified path name of the directory in which you originally installed WebSphere MQ Integrator Broker or WebSphere MQ Integrator.

DBMS	Change ...	To ...
Oracle	WMQIB\bin\ukor8dtc16.dll or WMQIB\bin\ukor8dtc18 .dll	WBIMB\bin\ukor8dtc20.dll
Sybase	WMQIB\bin\ukase16.dll or WMQIB\bin\ukase18.dll	WBIMB\bin\ukase20.dll

Migrating a Version 5.0 broker to Version 6.0 on z/OS

Migrate a broker to use the enhanced facilities available in a Version 6.0 broker domain.

To migrate a Version 5.0 broker to Version 6.0 on z/OS,:

1. Stop the Version 5.0 broker.
2. Back up the broker database tables.
3. Create a new broker PDSE.
4. Copy all broker JCL jobs from the Version 6.0 installed SBIPPROC and SBIPSAMP PDSEs to the new broker PDSE and customize them all. See Customizing the broker JCL for more information.

   When you customize the new Version 6.0 JCL for this migration, do not make the ++HOME++ JCL variable the same value as the existing Version 5.0 component directory, otherwise the previous ENVFILE will be overwritten when you submit the BIPGEN job. Set the ++HOME++ variable to the Started Task user ID home directory (for example, /u/mq01brk).

   a. Customize the BIPEDIT file using values that are defined in the broker's Version 5.0 mqsicompcif file. See "Equivalent z/OS customization variables in mqsicompcif and BIPEDIT" on page 74 for details of the customization variables.

   b. Copy any additional changes that you have made to the environment file, ENVFILE, and the ODBC initialization file, dsnaoini, to BIPBPROF and BIPDSNAO in the component data set. Submit the BIPGEN job to create the environment file ENVFILE.

   c. Customize and submit the BIPMGCMP job. This job migrates the registry, queue, and broker database. As part of the database migration, database tables are created, or deleted, and dropped, therefore you must you have the correct DB2 privileges. These privileges are the same as those required for running the mqsicreatebroker command.

5. Copy the started task job BIPBRKP to the procedures library. When you copy the started task, keep a second copy of the original in a safe place for backup purposes.
6. The verification program runs when you start the Version 6.0 broker.

When you have migrated the broker, migrate the rest of the components:

- Migrate the Message Brokers Toolkit
- Migrate the Configuration Manager
- Migrate the User Name Server

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Migrating a User Name Server from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0

Follow the instructions to migrate the User Name Server.

To migrate a User Name Server from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Event Broker Version 6.0, see the appropriate topic for your operating system:

- Distributed systems
- z/OS

Migrating a Version 5.0 User Name Server to Version 6.0 on distributed systems

Migrate a Version 5.0 User Name Server to Version 6.0.

The User Name Server component has not been changed for the Version 6.0 product. This component also retains no persistent state. Therefore you do not have to perform migration for this component. You can migrate the component, or you can delete and recreate it.

Before you start

If the User Name Server runs in a locale that is not listed in Locales, check that the code page is one of the supported code pages and that the locale is set up correctly.

To migrate the User Name Server:

1. Install Version 6.0 in a different location on the computer where WebSphere Business Integration Event Broker Version 5.0 is installed.
2. Stop the Version 5.0 User Name Server.
3. Set up the correct Version 6.0 command environment.

    - `Linux` `UNIX` On Linux and UNIX systems, run the environment profile mqsiprofile.

    - `Windows` On Windows, open a command console from the **Start** menu.
4. Enter the mqsimigratecomponents command to migrate the User Name Server. For example:

    ```
    mqsimigratecomponents usernameserver
    ```
5. Start the Version 6.0 User Name Server using the **mqsistart** command.

When you have migrated the User Name Server, migrate the rest of the components:

- Migrate the Message Brokers Toolkit
- Migrate the Configuration Manager
- Migrate the broker

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Migrating a Version 5.0 User Name Server to Version 6.0 on z/OS

Migrate a Version 5.0 User Name Server to Version 6.0.

To migrate a Version 5.0 User Name Server to Version 6.0 on z/OS, complete the following steps:

1. Stop the Version 5.0 User Name Server.
2. Create a new User Name Server PDSE.
3. Copy all User Name Server JCL from the Version 6.0 installed SBIPPROC and SBIPSAMP PDSEs to the new User Name Server PDSE and customize them all.

    When you customize the new Version 6.0 JCL for this migration, do not make the *++HOME++* JCL variable the same value as the existing Version 5.0 component directory, otherwise the previous ENVFILE will be overwritten when you submit the BIPGEN job. Set the *++HOME++* variable to the Started Task user ID home directory (for example, /u/mq01uns).

    a. Customize the BIPEDIT file using values that are defined in the User Name Server's Version 5.0 mqsicompcif file. "Equivalent z/OS customization variables in mqsicompcif and BIPEDIT" on page 74 lists the customization variables that are used in the Version 5.0 mqsicompcif file, and the equivalent variables in the Version 6.0 BIPEDIT file.

    b. Copy any additional changes that you have made to the environment file, ENVFILE, to BIPUPROF in the component data set. Submit the BIPGEN job to create the environment file, ENVFILE.

    c. Customize and submit the BIPMGCMP job.
4. Copy the started task JCL (BIPUNSP) to the procedures library. When you copy the started task, keep a second copy of the original in a safe place for backup purposes.
5. The verification program runs when you start the Version 6.0 User Name Server.

When you have migrated the User Name Server, ensure that you have migrated the rest of the components:

- Migrate the Message Brokers Toolkit
- Migrate the Configuration Manager
- Migrate the broker

When you have completed these tasks, see the post-migration tasks for information about tasks that you might want to perform after migration.

Post-migration tasks

After you have migrated from Version 5.0 to Version 6.0, finish setting up your environment.

Test the WebSphere Event Broker Version 6.0 broker resources and components to verify that you experience no loss or unexpected change of functionality. Some changes in behavior might be caused by defects that have been fixed between versions.

The following topics describe further tasks that you must do after migration:

- "Removing unwanted files and database tables after migration" on page 39
- "Setting up a command environment" on page 17

When you have done these tasks, you can delete the Version 5.0 components and remove the installed code. Access the Version 5.0 information center in the workbench for details of these tasks (or use the link in Related Links).

Removing unwanted files and database tables after migration

When you have completed migration from Version 5.0, you can remove resources from the computer.

Before you start

Make sure that the migration was successful and that you no longer need to retain the flexibility to use WebSphere Event Broker Version 5.0 code and resources.

To remove unwanted files and database tables after migration:

1. If you no longer need WebSphere Business Integration Event Broker Version 5.0, remove it from the computer. Refer to the WebSphere Business Integration Event Broker Version 5.0 information center for instructions about uninstalling the product.

2. If you no longer use prerequisite software products that are required by Version 5.0 but not for Version 6.0, remove them. For example, DB2 might no longer be required. For instructions, refer to the documentation for those products.

3. If you do not remove DB2, you might want to remove the DB2 database tables used by Configuration Manager Version 5.0.

   In Version 6.0, the Configuration Manager stores the domain configuration in an internal repository instead of an external database. Therefore you cannot use your Version 5.0 Configuration Manager DB2 database tables with the Version 6.0 Configuration Manager.

   To remove the Configuration Manager DB2 database tables, drop the following database tables by using the DB2 DROP TABLE command. If the Configuration Manager tables are in a DB2 database by themselves, drop the whole database by using the DB2 DROP DATABASE command.
   - CACLGROUPS
   - CBROKER
   - CBROKERCEG
   - CCOLLECTIVE
   - CCOLLECTIVECBROKER
   - CDELETE
   - CEG
   - CEGCMSGFLOW
   - CEGCMSGPROJECT
   - CLOG
   - CMSGFLOW
   - CMSGPROJECT
   - CNEIGHBOURS
   - COUTSTANDING
   - CPROXY
   - CPROXYCRESOURCE
   - CSUBSCRIBE
   - CTOPIC
   - CTOPICCTOPIC
   - CTOPOLOGY
   - CTRACE
   - CUUIDLOCKS

Setting up a command environment

After you have installed the product on any of the distributed systems, you must initialize the environment before you can use any runtime component or command.

Also complete this task if you have migrated to WebSphere Event BrokerVersion 6.0 from a previous version. A profile is provided to help you set up the environment.

Ensure that you use this environment each time you run one of the administrative commands, or start a component; for example, a broker.

For information on setting up your command and runtime environment on a Windows system, see Command environment: Windows platforms

For information on setting up your command and runtime environment on Linux and UNIX systems, see Command environment: Linux and UNIX systems

Restoring migrated components to previous versions

You can restore migrated components to previous versions of WebSphere Event Broker.

For some components, you restore them to the previous version and their original state; for other components, you restore them to the previous version, but keep any changes that you have made to them, such as deployed message flows.

You might want to restore components if you encounter a problem after you have migrated your broker domain to Version 6.0. By restoring components to the previous level, you can fix the problem with Version 6.0. You do not need to restore the entire domain; restore only those components that are causing a problem. For example, if a message flow does not work as expected in Version 6.0, restore that broker until you can resolve the problem.

You can restore components to a previous version only if they were migrated from that version originally. You cannot create a new Version 6.0 broker, then restore it to a previous level. Nor can you create a Version 2.1 broker, migrate it to Version 6.0, then restore it to Version 5.0.

The topics in this section explain how to restore components and resources that you have migrated from Version 5.0 or Version 2.1 products back to their original state:

- "Restoring components and resources to Version 5.0"
- "Restoring components and resources to Version 2.1" on page 43

Restoring components and resources to Version 5.0

If you have migrated from Version 5.0 to Version 6.0, you can restore your installation to Version 5.0, subject to certain restrictions.

Restrictions

- Source files in Message Brokers Toolkit Version 6.0 are maintained in a different format to that used in Version 5.0 and Version 5.1. The files are migrated to the new format when you save them using the Message Brokers Toolkit Version 6.0. After you have saved them, you can no longer use the resources in Message Brokers Toolkit Version 5.0 or

Version 5.1. For detailed information, see "Conditions for using migrated resources with previous versions of the Message Brokers Toolkit" on page 6.

- When you have restored a broker, you must redeploy the bar files that you had previously deployed at Version 6.0. If you do not redeploy, the deployed resources are not compatible with the Version 5.0 broker, and errors are generated when you restart the broker; always check logs to search for errors or warnings about message flows that the broker cannot start.

Restoring resources to Version 5.0

The following sections describe how to restore the Message Brokers Toolkit and your runtime components to Version 5.0.

Restoring the Message Brokers Toolkit to Version 5.0

To restore the Message Brokers Toolkit to Version 5.0:

1. Stop the Message Brokers Toolkit Version 6.0.
2. Restore the Version 5.0 workspace from the backup that you took before migration.

   If you did not create a backup of your Version 5.0 workspace, you cannot restore the Message Brokers Toolkit to this version.
3. Restart Message Brokers Toolkit Version 5.0 with your Version 5.0 workspace.

Any changes that you made to resources in the Message Brokers Toolkit Version 6.0 cannot be restored to Version 5.0.

Restoring runtime components to Version 5.0

Use the **-s** and **-t** parameters of the mqsimigratecomponents command to restore components from Version 6.0 to Version 5.0.

- Specify the installed level of Version 6.0 for the source version parameter (**-s**), for example `6.0.0.2` for Version 6.0, Fix Pack 2, or allow this to take the default value.
- Specify the appropriate level of Version 5.0 for the target version parameter (**-t**), for example `5.0.0.8` for Version 5.0, Fix Pack 8.

See the mqsimigratecomponents command for detailed information about these parameters and the format to use when specifying version numbers.

`z/OS` To restore z/OS runtime components to Version 5.0:

1. Submit the BIPMGCMP job to call the mqsimigratecomponents command, specifying the **-s** and **-t** parameters as described above.
2. Replace the started task JCL in USER.PROCLIB with the Version 5.0 backup that you took before migration.

Restoring a broker to Version 5.0

To restore a migrated broker to its Version 5.0 state:

1. Stop the Version 6.0 broker.
2. Restore the broker to Version 5.0 using the mqsimigratecomponents command, as shown in the following example:

   ```
   mqsimigratecomponents Broker -t 5.0.0.4
   ```

3. Reverse the changes that you made to the ODBC definitions when you migrated to Version 6.0.

- UNIX On Linux and UNIX, reset the ODBCINI environment variable to point to the previous version of the odbc.ini file.

- Windows On Windows, use the Control Panel to adjust the ODBC settings.

4. Restart the broker using a Version 5.0 command window.

Restoring a User Name Server to Version 5.0

The User Name Server is unchanged between Version 5.0 and Version 6.0.

If you choose to restore a User Name Server to Version 5.0, issue the mqsimigratecomponents command, as shown in the following example:

```
mqsimigratecomponents UserNameServer -t 5.0.0.4
```

Restoring a Configuration Manager to Version 5.0

When you migrate a Configuration Manager from Version 5.0 to Version 6.0, the DB2 database is neither changed nor deleted. Version 6.0 does not use DB2 for the configuration repository.

If you migrate from Version 6.0 back to Version 5.0, the original DB2 database is used again, so you must ensure that the database is available to the component. If you made any changes after migration to Version 6.0, these changes are not restored to Version 5.0. As a result, you will lose any domain changes that you made after migration to Version 6.0.

If you deployed any broker configuration changes after migration to Version 6.0, these changes are lost when you restore the Configuration Manager to Version 5.0. You must rebuild your brokers by deleting them and redeploying them to maintain a consistent state. Check carefully that your Configuration Manager works correctly before deploying for the first time after migration to Version 6.0.

If you changed the Configuration Manager queue manager during migration to Version 6.0, you must keep the new queue manager if you restore using the mqsimigratecomponents command. If the new queue manager is on an operating system other than Windows, you cannot use the mqsimigratecomponents command to restore; instead, you must restore from a backup.

The following example shows the command used to restore the Configuration Manager to Version 5.0 state on a Windows system:

```
mqsimigratecomponents ConfigMgr -t 5.0.0.4
```

Restoring components and resources to Version 2.1

Restore components and resources that you have migrated from Version 2.1 products back to their original version.

Restrictions

- You can restore resources to a Version 2.1 product only on operating systems that were supported by Version 2.1. You cannot restore resources to Version 2.1 on the following operating systems:
  - Linux (x86 platform)
  - Linux (zSeries® platform)
  - Linux (POWER™ platform)
  - Solaris (x86-64 platform)
- If you have used the following features in Version 6.0, restoring resources to Version 2.1 will result in loss of data:
  - Multicast
  - MQe nodes
  - Quality of Protection (QOP)
- When you restore a broker from WebSphere Event Broker Version 6.0 to Version 2.1, message sets are deleted, so you must redeploy them. You are reminded to do this by warning message BIP8688, which appears when you run the mqsimigratecomponents command.
- When you restore a broker from Version 6.0 to Version 2.1 on z/OS, retained publications are deleted.
- After migration to Version 6.0, if you have deployed any message flows that use new Version 6.0 functions, such as new nodes or new ESQL syntax, those message flows will not start if your broker is restored to Version 2.1. Other message flows in your broker that do not use new Version 6.0 functions will continue to run.
- Due to coexistence requirements, you can restore the Configuration Manager to Version 2.1 only if you have not migrated any brokers to Version 6.0. If you restore the Configuration Manager to Version 2.1, you must also restore the Message Brokers Toolkit to Version 2.1.

Restoring resources to Version 2.1

The following sections describe how to restore the Message Brokers Toolkit and runtime components to Version 2.1.

Restoring the Message Brokers Toolkit to Version 2.1

If you restore the Message Brokers Toolkit to Version 2.1, you must also restore the Configuration Manager to Version 2.1 at the same time. Any changes that you made after migration to Version 6.0 will be lost.

1. Migrate the Configuration Manager back to Version 2.1.
2. Restart the Version 2.1 Control Center.

Restoring runtime components to Version 2.1

Use the **-s** and **-t** parameters of the mqsimigratecomponents command to migrate components from Version 6.0 to Version 2.1. Specify Version 6.0 for the source version parameter (**-s**) and Version 2.1.0.8 for the target version parameter (**-t**). See the

mqsimigratecomponents command topic for detailed information
about these parameters and the format to use when specifying
version numbers.

z/OS To restore z/OS runtime components to Version 2.1,
perform the following steps:

1. Submit the BIPMGCMP job to call the mqsimigratecomponents
   command, specifying the **-s** and **-t** parameters as described
   above.
2. Replace the started task JCL in USER.PROCLIB with the
   Version 2.1 copy that you backed up.

Restoring a broker to Version 2.1

To restore a migrated broker to its Version 2.1 state, perform the following
steps:

1. Stop the Version 6.0 broker.
2. Restore the broker to Version 2.1 using the mqsimigratecomponents
   command, as shown in the following example:

   ```
   mqsimigratecomponents Broker -t 2.1.0.8
   ```

   Warning message BIP8688 might be displayed, warning you to
   redeploy your message sets.
3. Reverse the changes that you made to the ODBC definitions when you
   migrated to Version 6.0.

   - UNIX On Linux and UNIX, reset the ODBCINI environment
     variable to point to the previous version of the odbc.ini file.

   - Windows On Windows, use the Control Panel to adjust the ODBC
     settings.
4. Restart the broker.

Restoring a User Name Server to Version 2.1

The User Name Server has not changed between WebSphere MQ Event
Broker Version 2.1 and WebSphere Event Broker Version 6.0. To restore a
User Name Server to Version 2.1, issue the **mqsimigratecomponents**
command, as shown in the following example:

```
mqsimigratecomponents UserNameServer -t 2.1.0.8
```

Restoring a Configuration Manager to Version 2.1

If you restore the Configuration Manager to Version 2.1, you must also
restore the Message Brokers Toolkit and any brokers to Version 2.1.

The Version 6.0 Configuration Manager does not use the DB2 database that
was used by the Version 2.1 Configuration Manager; if you restore to
Version 2.1, the original database is used again. This means that any
configuration changes that you made after you migrated the Configuration
Manager to Version 6.0 will be lost. This can leave your domain in an
inconsistent state, forcing you to rebuild and redeploy all of your brokers
in order to regain a consistent state. After you have migrated your
Configuration Manager to Version 6.0, check carefully that the Message
Brokers Toolkit and Configuration Manager are working correctly before
you attempt any further configuration changes or before you migrate any
brokers.

If you have moved the Configuration Manager's queue manager during migration, you must keep the new queue manager when you restore, unless you restore your entire domain from a backup. If you have moved the Configuration Manager onto an operating system other than Windows, you cannot restore it to Version 2.1 unless you restore your domain from a backup.

Given these warnings, you can restore a Version 6.0 Configuration Manager to Version 2.1 using the command in the following example:

```
mqsimigratecomponents ConfigMgr -t 2.1.0.8
```

Your Version 6.0 Message Brokers Toolkit will no longer work; you must use a Version 2.1 Control Center instead.

Migrating publish/subscribe applications

If you are already using WebSphere MQ Publish/Subscribe, you can migrate your applications to use the publish/subscribe functions provided by WebSphere Event Broker.

Before you start this migration, check that you have the required level of WebSphere MQ for your WebSphere MQ Publish/Subscribe system. WebSphere Event Broker Version 6.0 requires at leastWebSphere MQ Version 5.3.

For more information, refer to the following topics:
- "Planning for migration"
- "Running two independent broker networks" on page 46
- "Creating and operating a heterogeneous network" on page 46
- "Migrating WebSphere MQ brokers" on page 52
- "Migrating a WebSphere MQ broker network" on page 55

Planning for migration

If you are already using WebSphere MQ Publish/Subscribe, you can migrate your applications to use the publish/subscribe functions provided by WebSphere Event Broker.

You can also migrate individual WebSphere MQ Publish/Subscribe brokers to create replacement WebSphere Event Broker brokers, with support for their client applications remaining intact.

These two possibilities offer you a number of advantages:
- Publications from within the WebSphere MQ Publish/Subscribe network can be targeted by WebSphere Event Broker subscribers. This includes messages originating in environments not yet supported by WebSphere Event Broker.
- Message flows can be created and deployed on WebSphere Event Broker brokers to:
  - Analyze the information that is flowing around your enterprise.
  - Create and execute additional business logic dependent upon the content of the publications.

– Consolidate the information within your enterprise in the form of new publications, that can then be republished as a series of additional topics available to both WebSphere Event Broker and WebSphere MQ Publish/Subscribe clients.

There are three possible scenarios for exploiting the two networks:

1. You can choose to have two independent broker networks, and therefore have two separate broker domains for publish/subscribe applications. See "Running two independent broker networks."

2. You can connect the two networks to allow publications and subscriptions to flow throughout the integrated network. Further details are provided in "Creating and operating a heterogeneous network."

3. You can selectively and gradually migrate individual brokers from WebSphere MQ Publish/Subscribe to WebSphere Event Broker. For more guidance on this option, see "Migrating WebSphere MQ brokers" on page 52 and "Migrating a WebSphere MQ broker network" on page 55.

Before you can make this choice, and create your migration plan, you must be aware of the differences between the two products. These differences are described in WebSphere MQ Publish/Subscribe.

Running two independent broker networks

If you already have an WebSphere MQ Publish/Subscribe broker network, you can continue to use this network unchanged. The introduction of WebSphere Event Broker Version 6.0 to your environment, and the creation of brokers in that broker domain, does not affect your WebSphere MQ Publish/Subscribe broker domain until you take specific action to connect the two networks.

If you want to run in this mode with two separate, independent networks, you do not have to take any specific actions. You can retain your existing WebSphere MQ Publish/Subscribe network, and install and configure a WebSphere Event Broker Version 6.0 network, without any interaction.

Your existing applications can continue to work unchanged. However, there can be no interchange of publications in this scenario.

You must be aware that a single queue manager cannot support both a WebSphere MQ Publish/Subscribe broker and a WebSphere Event Broker Version 6.0 broker. If you have brokers of both types on the same system, each broker must have its own dedicated queue manager.

You can implement this scenario while you assess the new product and the extra functions contained within the publish/subscribe support. It also lets you plan for the extent of integration or migration, or both, that you require, without affecting your current environment.

Creating and operating a heterogeneous network

You can integrate your existing WebSphere MQ Publish/Subscribe broker network with a WebSphere Event Broker broker network to create a mixed, heterogeneous network. This enables publications and subscriptions to be propagated through one logical network, made up of two or more physical networks. As a result, subscribers to the WebSphere MQ Publish/Subscribe brokers can target information being published to the WebSphere Event Broker broker network, and

subscribers to the WebSphere Event Broker brokers can target information being published to the WebSphere MQ Publish/Subscribe broker network.

In a heterogeneous network, any new stream that is created for a WebSphere MQ Publish/Subscribe broker is supported by any WebSphere MQ Publish/Subscribe brokers that are connected to it. However, the new stream is not supported by WebSphere MQ Publish/Subscribe brokers that are separated by a WebSphere Event Broker broker from the WebSphere MQ Publish/Subscribe broker where the new stream was created.

WebSphere MQ Publish/Subscribe brokers and WebSphere Event Broker brokers can be running on Windows or other platforms.

There are two ways in which a WebSphere Event Broker broker can be joined to the WebSphere MQ Publish/Subscribe network; it can be joined either as a leaf node (that is, as a child of an existing WebSphere MQ Publish/Subscribe broker) or as a parent node (that is, as the parent of an existing WebSphere MQ Publish/Subscribe broker).

Every WebSphere Event Broker broker that you integrate into an WebSphere MQ Publish/Subscribe network must have a minimum of two queues available:
- SYSTEM.BROKER.DEFAULT.STREAM

  This queue supports the default publication stream. You must create this queue on every broker. You must also create and deploy a message flow that services this stream queue.
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS

  This queue is used by the broker to communicate with neighboring WebSphere MQ Publish/Subscribe brokers. You must create this queue on every broker, however the message flow for this queue is created internally by the broker.

Adding a broker as a leaf node

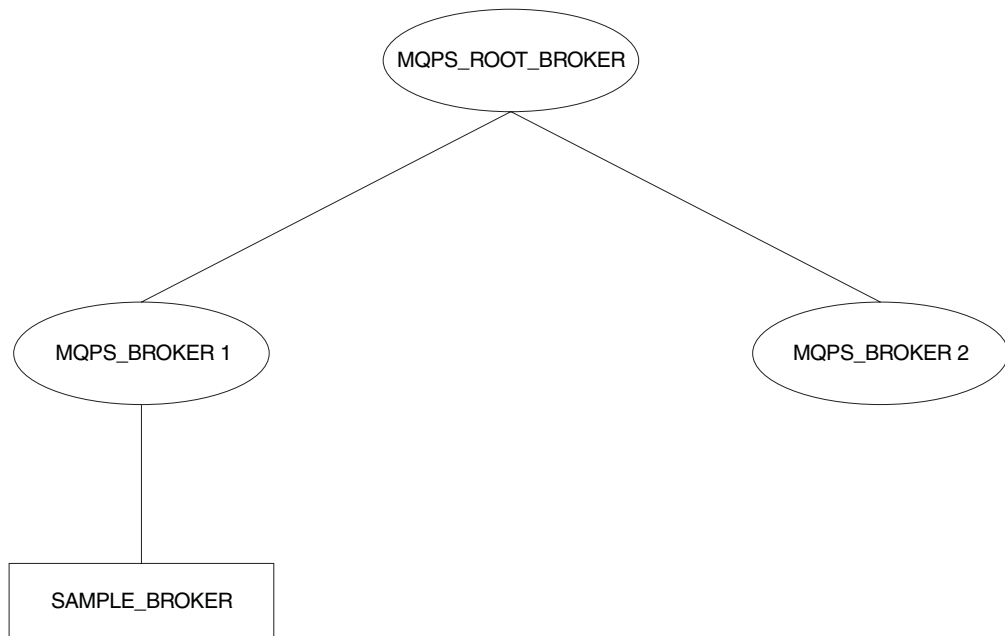The steps described in this topic assume that you are joining:
- A WebSphere Event Broker broker named WBRK_BROKER. When this broker was created, the same name was specified for the queue manager.
- A WebSphere MQ Publish/Subscribe broker network with a root broker MQPS_ROOT_BROKER, and two leaf brokers MQPS_BROKER1 and MQPS_BROKER2.

Substitute the names of your brokers for these examples wherever they are used.

All commands shown must be issued on the system on which the appropriate resource is defined. WebSphere MQ commands (for example, the command to define a queue) are shown in MQSC format. For more information about WebSphere MQ commands, refer to your WebSphere MQ documentation.

The following steps describe what you should do to add, as a leaf node within your WebSphere MQ Publish/Subscribe broker network, a WebSphere Event Broker broker that you have already created.

This is shown in the following diagram. WebSphere Event Broker broker **WBRK_BROKER** is joined to the WebSphere MQ Publish/Subscribe network, with broker **MQPS_BROKER1** as its parent broker.

```
                        ┌─────────────────────────┐
                        │    MQPS_ROOT_BROKER     │
                        └─────────────────────────┘
                         ╱                        ╲
                        ╱                          ╲
        ┌────────────────────┐            ┌────────────────────┐
        │   MQPS_BROKER 1    │            │   MQPS_BROKER 2    │
        └────────────────────┘            └────────────────────┘
                  │
                  │
        ┌────────────────────┐
        │   SAMPLE_BROKER    │
        └────────────────────┘
```

1. Ensure that the WebSphere Event Broker broker's default execution group is successfully deployed. This execution group is deployed the first time you deploy a newly created WebSphere Event Broker broker. You can check the status of both the execution group and the broker from the topology view in the workbench.

2. Define the queue required to support interbroker communications with WebSphere MQ Publish/Subscribe neighbors on the WebSphere Event Broker broker's queue manager:

   `define qlocal(SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS) noshare`

3. Stop the WebSphere Event Broker broker:

   `mqsistop WBRK_BROKER`

4. Restart the WebSphere Event Broker broker:

   `mqsistart WBRK_BROKER`

   When the WebSphere Event Broker broker is restarted, the presence of the interbroker queue (defined above) enables the broker to receive and process messages on this queue.

5. Create the resources required on the WebSphere Event Broker broker to support the default WebSphere MQ Publish/Subscribe stream:

   a. Create the default stream queue:

      `define qlocal(SYSTEM.BROKER.DEFAULT.STREAM) noshare`

   b. Create a message flow based on the supplied publish/subscribe message flow:

      1) Start up the workbench and select the designer view.

      2) If you have not already imported and saved the default message flows supplied, import these now. This enables you to reuse the default publish/subscribe flow here. Click **File → Import** and open the file, called `SamplesWorkspaceForImport`, in the `examples` subdirectory within the WebSphere Event Broker home directory. This might take a few minutes to complete.

         If you prefer, you can create your own message flow.

      3) Make a copy of the supplied message flow and rename it.

4) Check the properties of the nodes in the message flow. You must set the appropriate input (stream) queue property for the MQInput node. Check that the other properties of the nodes are set correctly for your requirements.

5) Finally, check in your changes and deploy the message flow to the default execution group of the broker WBRK_BROKER.

You can find full details of how to complete these steps in the online help for the workbench.

6. If you are using additional streams in the WebSphere MQ Publish/Subscribe network, you must also enable these on the WebSphere Event Broker broker. Although the WebSphere Event Broker broker is able to support all the streams of its WebSphere MQ Publish/Subscribe neighbors, you need only define queues, and define and deploy message flows, for those streams requested by WebSphere Event Broker subscriber clients.

   a. Create a local queue on the WebSphere Event Broker broker's queue manager for each stream on which messages are to be processed. For example:

   ```
   define qlocal(STREAM.X) noshare
   ```

   b. Create and deploy a message flow to read and process the WebSphere MQ Publish/Subscribe messages that are sent to each stream (publication) queue.

   You can use the supplied publish/subscribe message flow as the basis for each new message flow. Each MQInput node representing a non-default stream must have the property *implicitStreamNaming* set (this is the default setting).

7. Ensure that the WebSphere MQ Publish/Subscribe broker is running. If it is not, you can start it using the start command:

   ```
   strmqbrk MQPS_BROKER1
   ```

8. Ensure that the WebSphere MQ connection between the two brokers is enabled; you must start the listeners for the receiver channels, and you must then start the sender channels.

9. Join the WebSphere Event Broker broker to the WebSphere MQ Publish/Subscribe network as a child of the WebSphere MQ Publish/Subscribe broker:

   ```
   mqsijoinmqpubsub WBRK_BROKER -p MQPS_BROKER1
   ```

10. Verify the success of the join command to ensure that the WebSphere MQ Publish/Subscribe broker is an active neighbor:

   ```
   mqsilistmqpubsub WBRK_BROKER
   ```

   If the join command has completed successfully, you see a response to the list command that is similar to:

   ```
   BIP8090I: WebSphere MQ Publish/Subscribe neighbor WBRK_BROKER
        is active

   BIP8091I: Common stream    SYSTEM.BROKER.DEFAULT.STREAM
   BIP8091I: Common stream    STREAM.X
   ```

Adding a broker as a parent node

The steps described in this topic assume that you are joining:

- A WebSphere Event Broker broker named WBRK_BROKER. When this broker was created, the same name was specified for the queue manager.

- A WebSphere MQ Publish/Subscribe broker network with a root broker MQPS_ROOT_BROKER, and two leaf brokers MQPS_BROKER1 and MQPS_BROKER2.

Substitute the names of your brokers for these examples wherever they are used.

All commands shown must be issued on the system on which the appropriate resource is defined. WebSphere MQ commands (for example, the command to define a queue) are shown in MQSC format. For more information about WebSphere MQ commands, refer to your WebSphere MQ documentation.

The following steps describe what you should do to add, as a parent node within your WebSphere MQ Publish/Subscribe broker network, a WebSphere Event Broker broker that you have already created.

This is shown in the following diagram. WebSphere Event Broker broker **SAMPLE_BROKER** is joined to the WebSphere MQ Publish/Subscribe network as the new parent node (that is, as the parent of the original parent node **MQPS_ROOT_BROKER**).
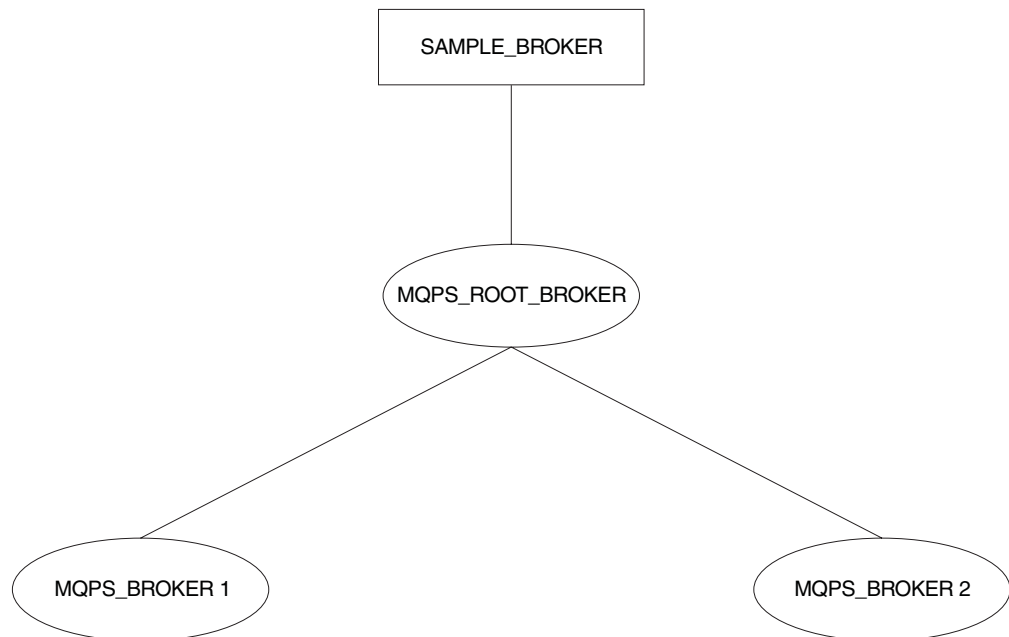
1. Ensure that the WebSphere Event Broker broker's default execution group is successfully deployed. This execution group is deployed the first time you deploy a newly created WebSphere Event Broker broker. You can check the status of both the execution group and the broker from the topology view in the workbench.
2. Define the queue required to support interbroker communications with WebSphere MQ Publish/Subscribe neighbors on the WebSphere Event Broker broker's queue manager:

   ```
   define qlocal(SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS) noshare
   ```
3. Stop the WebSphere Event Broker broker:

   ```
   mqsistop SAMPLE_BROKER
   ```
4. Restart the WebSphere Event Broker broker:

   ```
   mqsistart SAMPLE_BROKER
   ```

When the WebSphere Event Broker broker is restarted, it is enabled to receive and process messages on the interbroker queue.

5. Create the resources required on the WebSphere Event Broker broker to support the default WebSphere MQ Publish/Subscribe stream:

   a. Create the default stream queue:

      ```
      define qlocal(SYSTEM.BROKER.DEFAULT.STREAM) noshare
      ```

   b. Create a message flow for publish/subscribe, either your own, or one based on the supplied publish/subscribe message flow:

      1) Start up the workbench and select the designer view.

      2) Make a copy of the supplied message flow and rename it; you must import this default message flow before you can access and use it.

      3) Check the properties of the nodes in the message flow. You must set the appropriate input (stream) queue property for the MQInput node. Check that the other properties of the nodes are set correctly for your requirements.

      4) Finally, deploy the message flow to the default execution group of the broker SAMPLE_BROKER.

      You can find full details of how to complete these steps in the online help for the workbench.

6. If you are using additional streams in the WebSphere MQ Publish/Subscribe network, you must also enable these on the WebSphere Event Broker broker. Although the WebSphere Event Broker broker is able to support all the streams of its WebSphere MQ Publish/Subscribe neighbors, you need only define queues, and define and deploy message flows, for those streams requested by WebSphere Event Broker subscriber clients.

   a. Create a local queue on the WebSphere Event Broker broker's queue manager for each stream on which messages are to be processed. For example:

      ```
      define qlocal(STREAM.X) noshare
      ```

   b. Create and deploy a message flow to read and process the WebSphere MQ Publish/Subscribe messages that are sent to each stream (publication) queue.

      You can use the supplied publish/subscribe message flow as the basis for each new message flow. Each MQInput node representing a non-default stream must have the property *implicitStreamNaming* set.

7. Enter the following WebSphere MQ Publish/Subscribe command against the broker that is the current WebSphere MQ Publish/Subscribe parent broker, to terminate its activities:

   ```
   endmqbrk -c -m MQPS_ROOT_BROKER
   ```

   This requests a controlled shutdown (-c). When the shutdown has completed, the broker can be restarted. You can request an immediate shutdown, by specifying -iinstead of -c, if you need to force this shutdown to complete.

8. Ensure that the WebSphere MQ connection between the two brokers is active; you must start the listeners for the receiver channels, and then start the sender channels.

9. Enter the following WebSphere MQ Publish/Subscribe command against the current WebSphere MQ Publish/Subscribe parent broker to restart it:

   ```
   strmqbrk -m MQPS_ROOT_BROKER -p SAMPLE_BROKER
   ```

If the queue manager associated with the WebSphere Event Broker broker SAMPLE_BROKER has not been created with the same name as the broker, you must specify the queue manager name here after the -p flag, not the broker name.

10. Verify the success of the integration:

```
mqsilistmqpubsub SAMPLE_BROKER
```

If the WebSphere Event Broker broker has been integrated into the WebSphere MQ Publish/Subscribe network successfully, you see a response that is like the following:

```
BIP8090I: WebSphere MQ Publish/Subscribe neighbor SAMPLE_BROKER
   is active

BIP8091I: Common stream    SYSTEM.BROKER.DEFAULT.STREAM
BIP8091I: Common stream    STREAM.X
```

Deleting brokers

If you have a mixed broker network, you must take particular care to maintain the integrity of the network if you need to remove or delete a broker from the network:

- When you issue the mqsideletebroker command to delete a WebSphere Event Broker broker, the WebSphere MQ Publish/Subscribe brokers that are neighbors of this WebSphere Event Broker broker are not automatically informed of its deletion. Remove the WebSphere Event Broker broker from the network using the clear commands mqsiclearmqpubsub (at the WebSphere Event Broker broker) and clrmqbrk (at its WebSphere MQ Publish/Subscribe neighbors) before you delete it.

- If you delete a WebSphere Event Broker broker before you remove it from the network, and it has a parent WebSphere MQ Publish/Subscribe broker, the parent broker continues to attempt to send publication and subscription messages to it. You can correct this behavior by issuing the clrmqbrk command at the parent. For example, if you issue:

```
mqsideletebroker -m WBRK_CHILD_BROKER
```

while the WebSphere Event Broker broker is still known to its parent WebSphere MQ Publish/Subscribe broker, you can then issue the command

```
clrmqbrk -m MQPS_PARENT_BROKER -c WBRK_CHILD_BROKER
```

to the parent broker to clean up the network.

- When you issue the dltmqbrk command to delete an WebSphere MQ Publish/Subscribe broker that is a child of a WebSphere Event Broker broker, the WebSphere Event Broker broker receives notification of the deletion.

Therefore you do not have to issue the mqsiclearmqpubsub command to remove knowledge of the deleted child at the WebSphere Event Broker parent broker.

For example, if you want to delete the child broker MQPS_CHILD_BROKER you must issue the following single command:

```
dltmqbrk -m MQPS_CHILD_BROKER
```

You are prevented from deleting a WebSphere MQ Publish/Subscribe broker that is a parent of any broker; the dltmqbrk command fails.

Migrating WebSphere MQ brokers

Migrating WebSphere MQ brokers

When you plan the migration of one or more WebSphere MQ brokers, you must take account of the differences between WebSphere MQ Publish/Subscribe and

WebSphere Event Broker. This might mean that you have to make some changes to your applications, your topics, or both, before you start migration.

The information here tells you the steps you must take to migrate a single broker. "Migrating a WebSphere MQ broker network" on page 55 tells you how to migrate a WebSphere MQ broker network.

These steps result in the replacement of the WebSphere MQ brokers by WebSphere Event Broker brokers.

Each replacement WebSphere Event Broker broker must be created on the same queue manager as the WebSphere MQ broker that it replaces. Because the WebSphere MQ broker shares the same name as the queue manager that supports it, you must specify the WebSphere MQ broker name as the queue manager parameter on the mqsicreatebroker command (the -q flag).

Migration involves the transfer of the following state information from the WebSphere MQ broker to the WebSphere Event Broker broker:

- Subscriptions.

  All client subscriptions are exported from all streams except SYSTEM.BROKER.ADMIN.STREAM.

- Retained publications.

  All retained publications in MQRFH format are exported from all streams except SYSTEM.BROKER.ADMIN.STREAM.

- Local publishers.

  Registrations for all publishers that are producing local publications are exported from all streams except SYSTEM.BROKER.ADMIN.STREAM.

- Related brokers.

  If the broker is part of a multi-broker hierarchy, details of all its relations are exported. This includes the names of all streams that the broker to be migrated has in common with the relation.

- Streams.

  On WebSphere MQ Publish/Subscribe, streams are the queues from which publications are read by a broker.

This information is exported as a series of messages that are sent from the WebSphere MQ broker to its replacement. When migration is complete, the WebSphere MQ broker is deleted automatically, and cannot be recreated.

The workbench and migration

If you are migrating a WebSphere MQ broker, you cannot fully deploy it in your WebSphere Event Broker broker domain until migration has completed successfully. You should not deploy additional execution groups or message flows until after you have successfully migrated the WebSphere MQ Publish/Subscribe broker.

Use the *Broker Topology* editor to define the WebSphere Event Broker broker, and deploy the topology. Create an empty .bar file and drag it onto the default execution group. You are now ready to start the migration.

If migration fails, and you want to revert to your WebSphere MQ broker, you must delete the WebSphere Event Broker. See Deleting a broker.

Migrating a single broker

When you migrate a WebSphere MQ broker that is not part of a network, you are replacing it in the network and assigning all the function that was previously supported by that broker to a WebSphere Event Broker broker.

You must shutdown the WebSphere MQ broker before you start migration, and ensure that all applications that use the broker are also quiesced.

Preparing for the migration

Before you can migrate a broker, you need to do some preparation.

1. Identify the WebSphere MQ broker that you are going to migrate.

   The steps used here assume you have chosen the name WBRK_BROKER for your new WebSphere Event Broker broker, and that the WebSphere MQ broker you are migrating is currently hosted by the queue manager MQPS_BROKER1.

2. Back up the queue manager hosting the WebSphere MQ broker.

   Ensure that this backup is complete backup before you start the migration process. This allows you to retrieve the old WebSphere MQ broker after successful migration, if you should need to do so for any reason. The *WebSphere MQ System Administration* book describes this backup process.

3. Quiesce any applications that are registered with the broker.

   Any messages generated during the migration exercise are queued and could cause performance or capacity problems. Quiescing the applications as well as the broker ensures that publish/subscribe traffic is only generated when there is a broker ready to process it.

4. End your WebSphere MQ broker operation:

   ```
   endmqbrk MQPS_BROKER1
   ```

Preparing the replacement broker

You are now ready to work with the new broker.

1. Create a WebSphere Event Broker broker.

   You must create the new broker on the system on which the queue manager MQPS_BROKER1 is defined. You must select the migration option (flag -m) on the command.

   ```
   mqsicreatebroker WBRK_BROKER -q MQPS_BROKER1
   -i mqbroker -a sample -n WBRKBKDB -m
   ```

2. Start the new WebSphere Event Broker broker:

   ```
   mqsistart WBRK_BROKER
   ```

3. Configure the broker in the workbench.

   Create the new broker in the broker domain topology from the *Topology* view in the workbench. Save and deploy the topology. Create a default execution group and drag an empty .bar file onto the default execution group.

Migrating the WebSphere MQ broker

The new WebSphere Event Broker broker is ready to receive migration data for the WebSphere MQ broker that it replaces.

1. Migrate the WebSphere MQ broker function to the replacement WebSphere Event Broker broker by issuing the following command:

   ```
   migmqbrk -m MQPS_BROKER1
   ```

This command is supplied as part of the WebSphere MQ Publish/Subscribe package on the Web. You must ensure you have the latest level of this command, and the *WebSphere MQ Publish/Subscribe User's Guide* that describes its use.

The command retrieves the persistent information (subscriptions and retained publications) from the WebSphere MQ broker, and sends it in specially constructed messages to the queue SYSTEM.BROKER.INTERBROKER.QUEUE on the new WebSphere Event Broker broker.

The message flow that services this queue (deployed when you deployed the broker and its default execution group) receives these messages and records the information. When all the messages have been processed, the message flow is terminated and cannot be re-initialized.

The migration command can only be re-invoked if the whole process of migration has not completed successfully. If any error occurs, for any reason, the WebSphere MQ broker is recoverable and can be restarted. You can then continue to use it. The WebSphere Event Broker broker also exists, but has not recorded any migration information. You can delete and create this broker to restart the migration process.

If the whole process succeeds, the WebSphere MQ broker no longer exists and cannot be recovered.

You receive the following message on successful completion of migration:

```
WebSphere MQ broker has been successfully migrated
```

When you have successfully migrated all the WebSphere MQ brokers that you plan to migrate, delete or rename the file `strmqbrk.exe`. This prevents any WebSphere MQ brokers from starting accidentally.

Deploying the stream queues

The new WebSphere Event Broker broker is now set up to take over from the WebSphere MQ broker. You must create and deploy the message flows that it needs to activate the streams; you do not need to define the stream queues, because these are already defined to the queue manager. The queue definitions are not deleted when the migration takes place, and the same queue manager is used by the WebSphere MQ broker and the WebSphere Event Broker broker that has replaced it.

You can create the message flows you need by following these steps:

1. Start the workbench and select the message flow view.
2. For each stream, including the default stream:
   a. Build a basic publish/subscribe message flow by copying and renaming the supplied publish/subscribe message flow.
   b. Check the properties of the nodes in each message flow that you create.

      You must set the input queue name (the stream queue) property in the input node. You must also set the implicitStreamNaming property for every non-default stream queue input node.
   c. Finally, assign the message flow to an execution group of the broker WBRK_BROKER, check in your changes, and deploy the broker.

Migrating a WebSphere MQ broker network

The procedure that you must follow to migrate a WebSphere MQ broker that is part of a multi-broker network is basically the same as that needed to migrate a single broker.

Before you start the migration, you must consider:
- The order in which you migrate the brokers.
- The place of each broker in the network, and the relationships it has with its neighbors.
- The use of collectives in the WebSphere Event Broker network.

Refer to "Planning for migration" on page 45 for more information.

The following sequence of figures illustrates the migration of a network of three brokers. The actions taken to migrate the network assumes that the three brokers are migrated one at a time, and that all three are to be grouped in a single collective in the WebSphere Event Broker broker domain.

The WebSphere MQ broker network that is to be migrated has three brokers, the root (NEWYORK) and two children (LONDON and TOKYO).
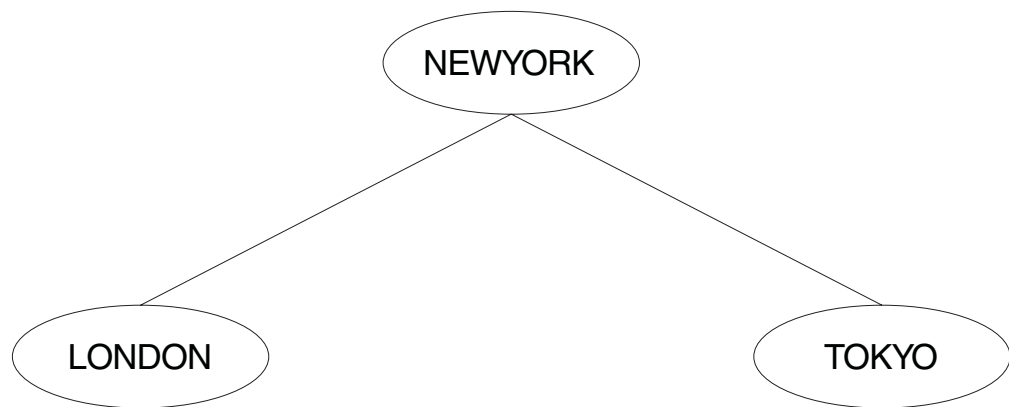
These brokers do not have to be migrated in any particular order. This example shows the migration being done in the following order:
1. LONDON
2. NEWYORK
3. TOKYO

The migration is completed in a number of separate steps. Each step is best taken when network traffic is low (for example, at weekends). The whole migration is planned in three stages:

Stage 1: migration of the LONDON broker

The steps that you need to take to migrating a single broker within a network are exactly the same as those that you need to take to migrate a standalone WebSphere MQ broker. See "Migrating WebSphere MQ brokers" on page 52.
1. Quiesce all client applications at both the LONDON and NEWYORK brokers. This ensures that no publications are missed by any subscribers while the topology is being changed.
2. Quiesce all other brokers in the network (in this example, the TOKYO broker). This guarantees that no publications are delivered while the topology is being changed.

After the migration of the LONDON broker you have a mixed network, consisting of two WebSphere MQ brokers (NEWYORK and TOKYO) and one WebSphere

Event Broker broker:

```
                        ╭───────────╮
                        │  NEWYORK  │
                        ╰───────────╯
                       ╱             ╲
                      ╱               ╲
           ┌──────────┐          ╭─────────╮
           │  LONDON  │          │  TOKYO  │
           └──────────┘          ╰─────────╯
```
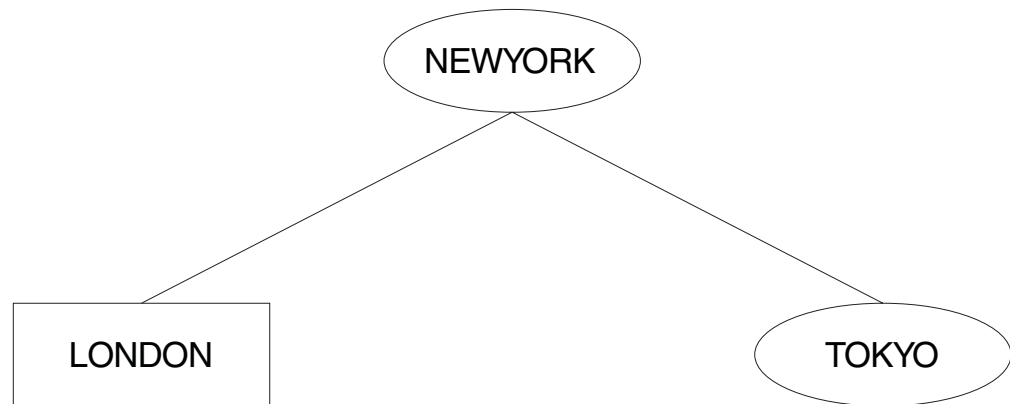
The connection between the LONDON and NEWYORK brokers is a WebSphere MQ connection. The workbench only recognizes WebSphere Event Broker brokers, and therefore only LONDON has been defined to it. A WebSphere Event Broker connection cannot be created at this stage.

This mixed network is in a perfectly valid state. It can remain in this state until you are ready to do the next stage of the migration.

Stage 2: migration of the NEWYORK broker

Follow the step-by-step procedure for migrating a single broker for broker NEWYORK. This is described in "Migrating WebSphere MQ brokers" on page 52.

1. Quiesce all client applications at all brokers to which NEWYORK is a neighbor; in this network, LONDON and TOKYO. This ensures that no publications are missed by any subscribers while the topology is being changed.

2. Quiesce all the brokers in the network. This guarantees that no publications are delivered while the topology is being changed.

The network now contains two WebSphere Event Broker brokers (LONDON and NEWYORK), and one WebSphere MQ broker (TOKYO):

```
                        ┌───────────┐
                        │  NEWYORK  │
                        └───────────┘
                       ╱             ╲
                      ╱               ╲
           ┌──────────┐          ╭─────────╮
           │  LONDON  │          │  TOKYO  │
           └──────────┘          ╰─────────╯
```
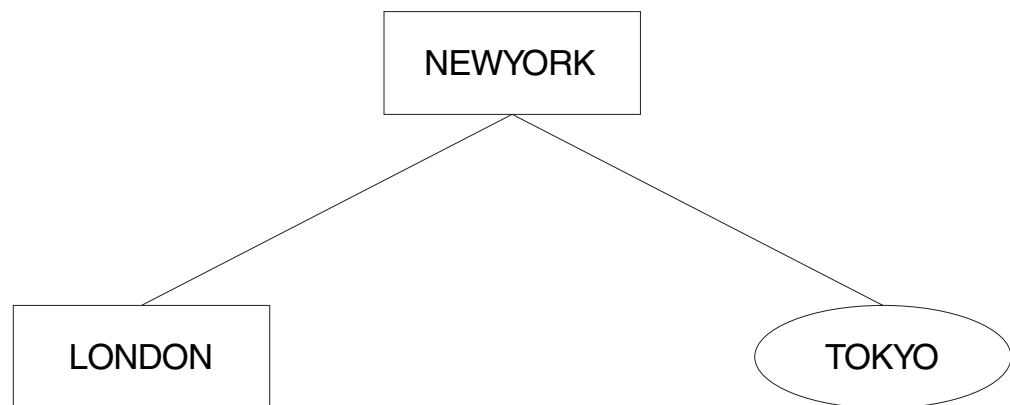
The LONDON and NEWYORK brokers are still connected by a WebSphere MQ connection. They can remain connected in this way for as long as necessary.

However, to develop applications that use the functions provided by WebSphere Event Broker you must join the two WebSphere Event Broker brokers together using the workbench.

The connection can be upgraded to a WebSphere Event Broker connection by first removing the original WebSphere MQ connection between LONDON and NEWYORK.

To remove this connection, issue the WebSphere Event Broker command mqsiclearmqpubsub at both brokers:

```
mqsiclearmqpubsub NEWYORK -n LONDON
mqsiclearmqpubsub LONDON -n NEWYORK
```

The network now looks like this:
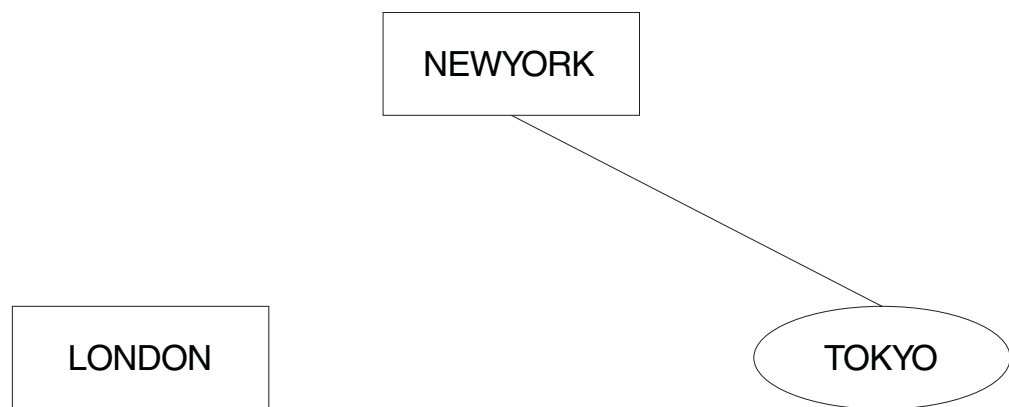
Now use the workbench to define the relationship between the two brokers, LONDON and NEWYORK. Both brokers are already defined, but the collective to which they are to be assigned is not. You can define this collective from the *Topology* view, and assign the two brokers to it. All brokers in a collective are assumed to be connected, so you do not have to make those connections using the workbench.

The new topology can now be deployed. The connection between LONDON and NEWYORK is now implemented using WebSphere Event Broker functions. The network is now:

The two brokers, LONDON and NEWYORK, are no longer in a parent-child relationship but are neighbors within a collective. The topology of the WebSphere Event Broker network is not based on a hierarchical structure as was the WebSphere MQ network.

Now that LONDON and NEWYORK form a collective, there is no root node left in the WebSphere MQ network. NEWYORK is the gateway between the WebSphere MQ broker (TOKYO) and the WebSphere Event Broker collective of brokers.

Stage 3: migration of the TOKYO broker

The final WebSphere MQ broker, TOKYO, is now ready to be migrated. Follow the procedure described in "Migrating WebSphere MQ brokers" on page 52.

The network is now:

The WebSphere MQ connection between TOKYO and NEWYORK can now be broken. This is done by using the following commands:

```
mqsiclearmqpubsub NEWYORK -n TOKYO
mqsiclearmqpubsub TOKYO -n NEWYORK
```

Now use the workbench to add the TOKYO broker to the WebSphere Event Broker network, and to the collective. The operation of a collective requires that all brokers have direct physical connections with each other (via WebSphere MQ).

Before the topology of the new WebSphere Event Broker network can be deployed, a WebSphere MQ connection between LONDON and TOKYO is required. A series of WebSphere MQ commands must be invoked to define the channels and transmission queues that support two-way traffic.

When you have completed the migration of all the brokers in the collective, you have removed the single point of failure at the NEWYORK broker. Subscribers on the LONDON broker can receive publications from the TOKYO broker even when the NEWYORK broker is not running.

Before migration, traffic between brokers was always routed through NEWYORK, the root node, which was therefore the single point of failure.

For further details of connecting brokers to each other, see Configuring WebSphere Event Broker. For more general information about distributed WebSphere MQ networks, refer to *WebSphere MQ Intercommunication*.

When all migration and associated tasks have been completed, the network comprises a single collective, containing three WebSphere Event Broker brokers connected as equals.
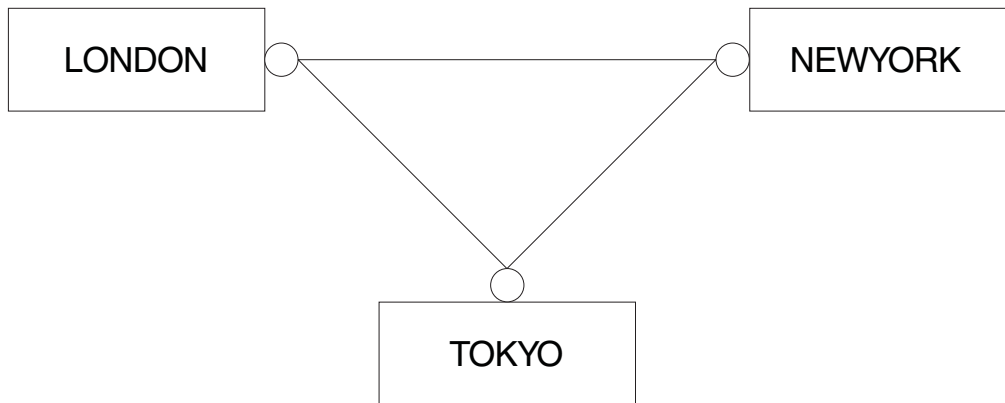
A network of migrated brokers

The following diagram shows a mixed network of WebSphere Event Broker and WebSphere MQ brokers. Brokers NEWYORK, LONDON, and TOKYO have been migrated to form a WebSphere Event Broker collective. All the other brokers remain as WebSphere MQ brokers.
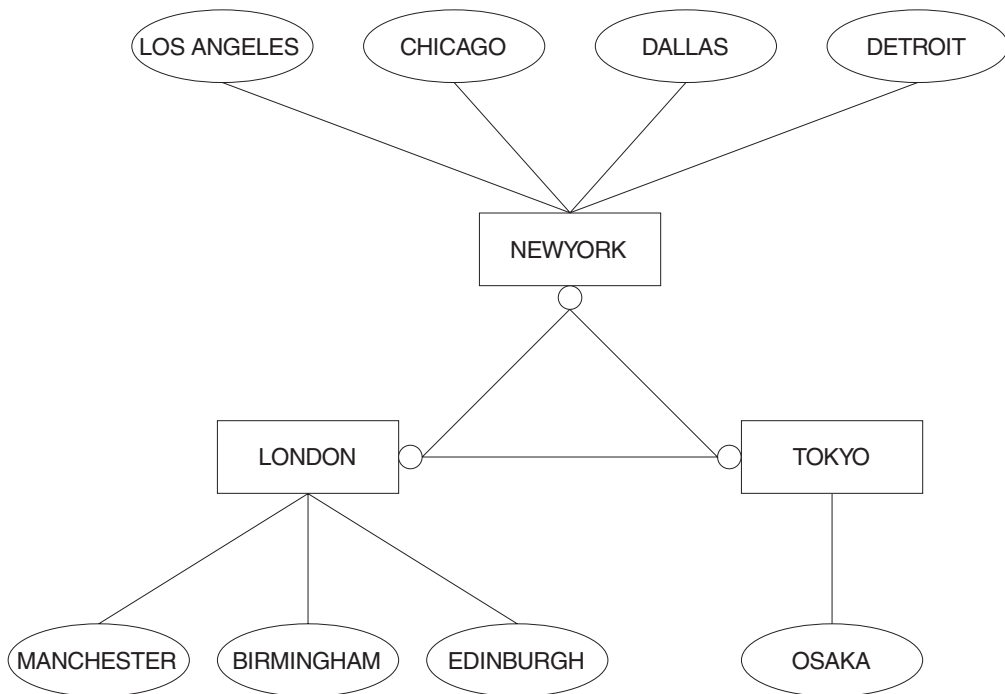
Part 2. Reference

Migration and upgrade

Consider the factors involved in the migration of components and resources from Version 2.1 or Version 5.0 to Version 6.0.

This section contains the following topics:

- "Supported migration and upgrade paths"
- "Message flow migration notes" on page 64
- "Equivalent z/OS customization variables in mqsicompcif and BIPEDIT" on page 74
- "Assignments configuration data in an export file" on page 75
- "Migration glossary" on page 78

Supported migration and upgrade paths

Migration and upgrade paths are provided to Version 6.0 products.

From ...	You can migrate to ...
MQSeries Integrator Version 2.0.2	• Migration from MQSeries Integrator Version 2.0.2, or an earlier release, is not supported.
WebSphere MQ Event Broker Version 2.1 (at any service level)	• WebSphere Event Broker Version 6.0 • WebSphere Message Broker Version 6.0
WebSphere MQ Integrator Broker Version 2.1 (at CSD02 or later)[1]	• WebSphere Message Broker Version 6.0
WebSphere MQ Integrator Version 2.1 (at CSD02 or later)[1]	• WebSphere Message Broker Version 6.0[2] • WebSphere Message Broker with Rules and Formatter Extension Version 6.0
WebSphere MQ Event Broker Version 2.1 (at any service level) and WebSphere MQ Integrator Broker Version 2.1[3] (at CSD02 or later)[1]	• WebSphere Message Broker Version 6.0
WebSphere MQ Event Broker Version 2.1 (at any service level) and WebSphere MQ Integrator Version 2.1[3] (at CSD02 or later)[1]	• WebSphere Message Broker Version 6.0[2] • WebSphere Message Broker with Rules and Formatter Extension Version 6.0
WebSphere Business Integration Event Broker Version 5.0	• WebSphere Event Broker Version 6.0 • WebSphere Message Broker Version 6.0
WebSphere Business Integration Message Broker Version 5.0	• WebSphere Message Broker Version 6.0
WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0	• WebSphere Message Broker Version 6.0 [4] • WebSphere Message Broker with Rules and Formatter Extension Version 6.0
WebSphere MQ Publish/Subscribe	• WebSphere Event Broker Version 6.0 • WebSphere Message Broker Version 6.0
WebSphere Event Broker Version 6.0	• WebSphere Message Broker Version 6.0

Notes:

1. If you have applied CSD02, CSD03, CSD04, or CSD05 to Version 2.1, you must apply an additional APAR (IY45459) in order for migration to be successful. APAR IY45459 is included in CSD06.

2. This migration path is possible only if you have not used the New Era of Networks Rules and Formatter support in WebSphere MQ Integrator Version 2.1, or you do not want to continue using it after migration.

3. Two broker domains are merged into one broker domain during migration.

4. This migration path is possible only if you have not used the Rules and Formatter function in WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0, or you do not want to continue using it after migration.

Message flow migration notes

This section provides reference information to help you to migrate message flows to WebSphere Event Broker Version 6.0.

It contains the following topics:
- "Migrating message flows from Version 2.1"
- "Migrating a user-defined node from Version 5.0 or Version 5.1" on page 65
- "Migrating a message flow that contains WebSphere MQ Everyplace nodes" on page 67

If you are using the Message Brokers Toolkit Version 5.1, replace all references in these topics to "Version 5.0" with "Version 5.1".

Migrating message flows from Version 2.1

Use the **mqsimigratemsgflows** command to migrate message flows from Version 2.1 to Version 6.0. **It is not necessary to use this command when migrating from Version 5.0 to Version 6.0.**

This topic contains the following sections:
- "Conditions for using the mqsimigratemsgflows command"
- "Promoted property name" on page 65
- "Converting Version 2.1 names that are not valid" on page 65

Conditions for using the mqsimigratemsgflows command
- Export message flows from Version 2.1 to an export file. You can use SupportPac utilities to create the files, and these exported files are the input to the migration process.
- Message flows that are found in multiple export files are migrated each time that they occur. When migrating a message flow, any previous flow with the same name is overwritten without warning.
- Message flows and their related subflows should be exported to the same file. This is the default Version 2.1 export behavior. This helps you to identify and resolve references to the other message flows and subflows, and to rename objects consistently. The migration uses the name of the message flow rather than the UUID to link flows and subflows and to name the message flow file.

- If a message flow refers to a subflow that cannot be located, but appears to be in the project, it is possible that it has been migrated separately, and was referenced internally in an inconsistent manner due to renaming. To correct the situation, export the message flows again in the same export file.
- When migration is complete, open the workbench. If the project is closed, open it; if the project is open, refresh it and clean all the projects by clicking `Project -> Clean`.

Promoted property name

In Version 2.1, when a promoted property is created through the drag-and-drop process, the property name (xmi.label) is set to be the translation of the attribute name. The original attribute name must not contain spaces otherwise it is rejected by the broker. However, promoted attributes are never sent to the broker, so they might have contained spaces in Version 2.1.

When the message flow is migrated, the original name is lost and only the translation is kept. The promoted attribute can override several attributes, so the original name must correspond to the translated name.

The solution is to generate a suitable attribute name by replacing spaces or other offending characters with the unicode representation. The *propertyName* attribute of the propertyDescriptor is set to `key=Property.`*<the translated attribute name>*. The UI returns *<the translated attribute name>*.

However, migrated message flows have not retained the attribute system name, only the translated name. It is therefore difficult or impossible to locate the original attribute.

Converting Version 2.1 names that are not valid

Message flows and properties can contain names that are not valid in Version 6.0. If this situation arises, the following transformation occurs. Each offending character is replaced with a series of characters representing its unicode code point. For example, an exclamation mark ("!") is replaced with X0026. This is explained in the report file that is generated.

This transformation is deterministic. If a message flow is migrated on another occasion, and the message flow refers to a flow with a character that is not valid, both names are transformed in the same way.

These transformations do not result in conflicting names except in extremely rare circumstances. A conflict might occur because a Unicode code point sequence occurs in a name precisely where the corresponding character occurs in another name, which is otherwise identical. In this case, rename one of these message flows or properties and re-export the flows. Select a new name that does not contain a Unicode code point sequence ('Xnnnn') and rename the message flow in the Control Center before you migrate. Never rename a .msgflow file in the file system; always use the Control Center or the workbench to perform renaming tasks.

Migrating a user-defined node from Version 5.0 or Version 5.1

Migrate your user-defined nodes to Version 6.0.

Migrating a user-defined node from Version 5.0

To migrate a user-defined node from Version 5.0 to Version 6.0, you must migrate both toolkit and runtime components.

Complete the following steps to migrate a user-defined node:

1. "Migrating the Message Brokers Toolkit representation of the user-defined node"
2. "Migrating the user-defined node runtime code"

Migrating the Message Brokers Toolkit representation of the user-defined node:

To migrate from the Message Brokers Toolkit Version 5.1 to the Message Brokers Toolkit Version 6.0:

1. Import the user-defined node project into the Message Brokers Toolkit Version 6.0.
2. Select your user-defined node project in the Package Explorer, and click **Project** → **Clean Project**.

To migrate from the Message Brokers Toolkit Version 5.0 to the Message Brokers Toolkit Version 6.0:

1. Import the user-defined node project into the Message Brokers Toolkit Version 6.0.
2. Select your user-defined node project in the Package Explorer, and click **Project** → **Clean Project**.
3. Modify the `<requires>` element in the plugin.xml file in the user-defined node project root to match the following content:

   ```
   <requires>
           <import match="greaterOrEqual" plugin="com.ibm.etools.mft.api" version="6.0.0"/>
   </requires>
   ```
4. Modify the extension point for "org.eclipse.help.contexts" in the same plugin.xml file to match the following content:

   ```
   <extension point="org.eclipse.help.contexts">
    <contexts file="HelpContexts.xml"/>
   </extension>
   ```

When you have migrated your user-defined nodes, you do not need to migrate any message flows that contain the user-defined node.

Now complete the step "Migrating the user-defined node runtime code."

Migrating the user-defined node runtime code:
Before you start this step, you must have migrated the Message Brokers Toolkit representation of the user-defined node.

To migrate the user-defined node runtime code:

1. Store a copy of your compiled or packaged user-defined extension file on every broker system on which you intend to use it. Save the .lil or .jar file in the directory that you specified on the mqsicreatebroker or mqsichangebroker command for each Version 6.0 broker. (On Version 5.0 broker systems, the .lil or .jar file was saved in the installation directory.)
   * If you are migrating a Java™ user-defined node, you can build the user-defined extension file once and distribute it to each of your systems.

- If you are migrating a C user-defined node, and all of your brokers are of the same machine type, you can build the user-defined extension file once and distribute it to each of your systems.
- If you are migrating a C user-defined node and you have a cluster that consists of various machine types, for example one AIX, one Solaris, and one Windows broker, you must build the files separately on each machine type.

2. For C user-defined extensions, also store the .pdb file that corresponds to the .lil file, in the specified directory. The .pdb file provides symbolic information that is used in stack diagnostic information in the event of access violations or other software malfunctions.

3. Stop and start each broker to ensure that the existence of a new file is detected.

   A broker restart is not required in the following two scenarios:
   - If you have created an execution group in the Message Brokers Toolkit, and you have not yet deployed to it, you can add the .lil, .pdb, and .jar file to your chosen directory.
   - If you have already deployed to the execution group that you want to use, add the .lil, .pdb, or .jar file to your chosen directory, then use the mqsireload command to restart the group. You cannot overwrite an existing file on the Windows operating system when the broker is running because of the file lock that is put in place by the operating system.

   Use these two approaches with caution because any execution group that is connected to the same broker will also detect the new .lil, .pdb, and .jar files when that execution group is restarted, or when something is first deployed to that execution group. By using the more conventional way of restarting the broker, you ensure that anyone with an interest in a particular execution group is made aware that recent changes have been made to the broker.

   These two situations assume that you have already completed the previous step, and have therefore used either the mqsichangebroker command or the mqsicreatebroker command to notify the broker of the directory in which the user-defined extension files have been placed.

   When you have installed a user-defined node, it is referred to by its schema and name, just like a message flow.

Migrating a message flow that contains WebSphere MQ Everyplace nodes

Support for WebSphere MQ Everyplace in WebSphere Event Broker Version 6.0 differs from that provided in WebSphere Business Integration Event Broker Version 5.0, and you must update message flows that make use of the support.

This topic summarizes the changes, and the subsequent topics describe the situation in more detail:
1. Migrating an MQe message flow (this topic)
2. "Designing MQe connections" on page 68
3. "Deploying an MQe message flow" on page 70
4. "Configuring after MQe message flow deployment" on page 71

Version 5.0 configuration
- You can use either of these connection configurations:
  - MQe device client <--*connects to* --> MQe gateway <--*connects to* --> broker (using MQInput and MQOutput nodes)
  - MQe device client <--*connects to* --> broker (using MQeInput and MQeOutput nodes)

- The MQe code level in the MQe nodes in the broker is at a fixed version, and you cannot update it independently.

Version 6.0 configuration

Using message flows that contain MQeInput and MQeOutput nodes in Version 6.0 is deprecated. The behavior that is described here is intended only for when you are deploying from Version 6.0 to a previous version, and to provide a route for migration. Redesign your flows to remove the MQe nodes and replace them with MQ nodes that are configured to your own specifications and coordinated with your MQe gateway configuration.

- You must install WebSphere MQ Everyplace separately. The WebSphere MQ Everyplace code is now all contained in that separate installation and it can be upgraded independently to newer versions.
- The MQeInput and MQeOutput nodes are in the workbench for compatibility only, and are not implemented within the broker component.
- You can use only this connection configuration:
  - MQe device client <--*connects to* --> MQe gateway <--*connects to* --> broker (using MQInput and MQOutput nodes)
- You can migrate your existing MQe message flows unchanged to Version 6.0, but you must configure the MQe gateway according to the modified implementation:
  - The bar file generated is identical to that created in Version 5.0.
  - When you deploy to a Version 5.0 broker, the flow works just as before.
  - When you deploy to a Version 6.0 broker, the broker interprets the bar file differently, and changes the MQeInput and MQeOutput nodes internally into MQInput and MQOutput nodes. The broker also ignores some MQe node properties, and reinterprets others.
- If you are using MQeOutput nodes with the Destination Mode property set to Destination List, when you migrate your MQe flows to a Version 6.0 broker, change the Destination Mode property to Reply to Queue, and then redeploy the modified flows. You can use the Message Brokers Toolkit at either Version 6.0, or at Version 5.0 (or Version 5.1) to perform the redeployment.

If you want to use WebSphere MQ Everyplace connections with Version 6.0, see the subsequent topics for more details, starting with "Designing MQe connections."

For help with configuring WebSphere MQ Everyplace, see the documentation supplied with that product.

Designing MQe connections

Design the connections used by your MQe message flows when you migrate from Version 5.0 to Version 6.0.

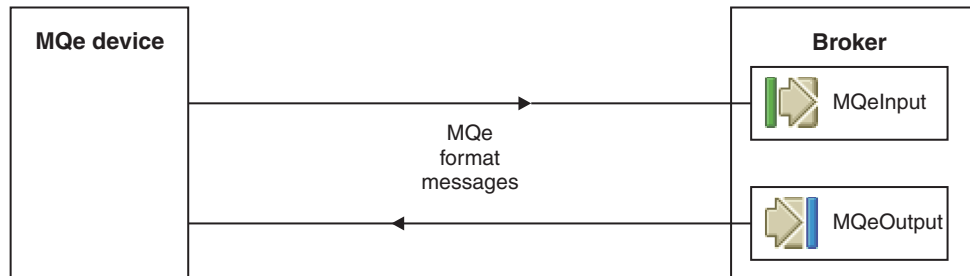This topic describes one in a sequence of migration tasks; complete these tasks in the order shown:
1. "Migrating a message flow that contains WebSphere MQ Everyplace nodes" on page 67
2. Designing MQe connections (this topic)
3. "Deploying an MQe message flow" on page 70
4. "Configuring after MQe message flow deployment" on page 71

This topic describes the connection options between WebSphere MQ Everyplace device clients and WebSphere Event Broker.
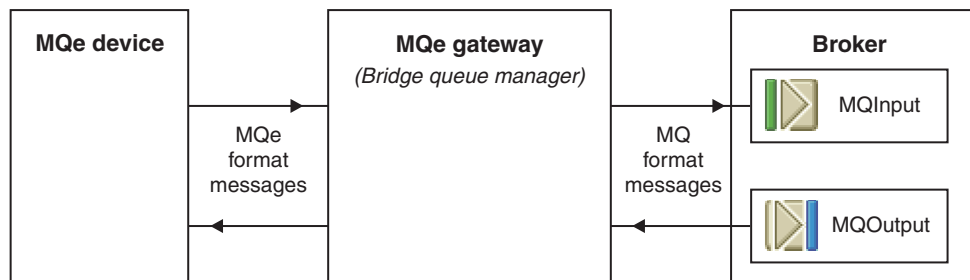
The following two diagrams represent how the connections are made; the table following the diagrams provides the details.

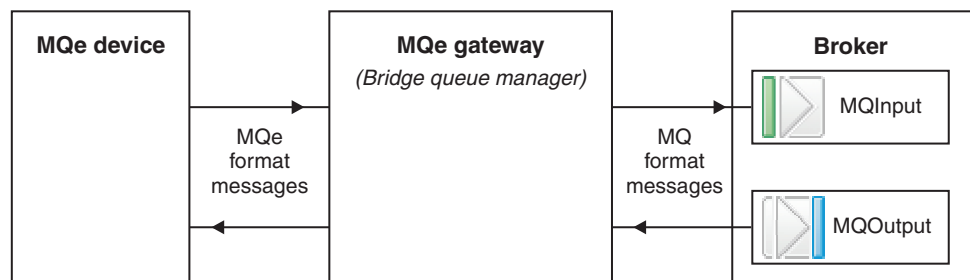1. The connection options for Version 5.0:
   - Direct:

MQe device → MQe format messages → **Broker** (MQeInput, MQeOutput)

   - Through an MQe gateway:

MQe device → MQe format messages → **MQe gateway** *(Bridge queue manager)* → MQ format messages → **Broker** (MQInput, MQOutput)

2. The connection option for Version 6.0:
   - Through an MQe gateway:

MQe device → MQe format messages → **MQe gateway** *(Bridge queue manager)* → MQ format messages → **Broker** (MQInput, MQOutput)

The following table lists connection options between WebSphere MQ Everyplace device clients and WebSphere Event Broker.

Connection considerations	Version 5.0	Version 6.0
MQe devices connect directly to the broker	Yes	No
MQe devices connect to an MQe gateway that connects to the broker	Yes	Yes
The format of messages into and out of the broker is:	MQe format or MQ format	MQ format only

The next task in this migration sequence is "Deploying an MQe message flow."

For help with configuring WebSphere MQ Everyplace, see the documentation supplied with that product.

Deploying an MQe message flow

Deploy your MQe message flows after you have migrated them from Version 5.0 to Version 6.0.

This topic describes one in a sequence of migration tasks; complete these tasks in the order shown:
1. "Migrating a message flow that contains WebSphere MQ Everyplace nodes" on page 67
2. "Designing MQe connections" on page 68
3. Deploying an MQe message flow (this topic)
4. "Configuring after MQe message flow deployment" on page 71

Using message flows that contain MQeInput and MQeOutput nodes in Version 6.0 is deprecated. The behavior that is described here is intended only for when you are deploying from Version 6.0 to a previous version, and to provide a route for migration. Redesign your flows to remove the MQe nodes and replace them with MQ nodes that are configured to your own specifications and coordinated with your MQe gateway configuration.

The following table describes the options when you deploy a message flow that contains MQeInput and MQeOutput nodes.

Deploy from	Deploy to	Result
Version 5.0	Version 5.0	Works as before
Version 5.0	Version 6.0	Not supported. Import your existing message flow into the Message Brokers Toolkit Version 6.0 and deploy from there.[1]
Version 6.0	Version 5.0	Works as before.[2]
Version 6.0	Version 6.0	Works when you set up an MQe gateway that you have configured appropriately. When the broker receives the bar file, the MQeInput and MQeOutput nodes in the message flow are converted into MQInput and MQOutput nodes that are re-configured as described below.

Notes:

1. When you save the flow in Version 6.0, you see an error message saying that the use of MQe nodes is deprecated. You can ignore the error and continue to save and deploy.

2. The bar files generated by Version 6.0 contain XML that is identical to that produced by Version 5.0.

The following list describes what happens to all the MQe node properties when the node is converted into an MQ node. (For a summary of the key properties, see "Configuring after MQe message flow deployment").

MQeInput and MQeOutput node properties panels	Converted MQ node properties
General panel properties • Trace • Trace Filename • Use Config File • Config Filename • Queue Manager Name • Queue Name	All discarded except: • Queue Manager Name • Queue Name Do your tracing in the normal way for WebSphere Event Broker, as described in Using trace. When this flow is traced, the MQ nodes are seen, not the MQe nodes in the message flow in the deployed bar file that have been converted.
Registry panel properties • Registry type • Directory • PIN • Certificate Request PIN • Keyring Password • Certificate Host • Certificate Port	All discarded These properties concern MQe registry and security, and you configure these characteristics on your MQe gateway.
Listener panel properties • Listener Type • Host Name • Port • Time Interval	All discarded You configure these properties on your MQe gateway and your broker queue manager.

To use such a deployment, create a suitable configuration on the broker queue manager for the converted nodes to work, as described in the next topic, "Configuring after MQe message flow deployment."

For help with configuring WebSphere MQ Everyplace, see the documentation supplied with that product.

Configuring after MQe message flow deployment

Create a configuration on the broker queue manager to handle the MQe message flows that you have migrated from Version 5.0 to Version 6.0.

This topic describes one in a sequence of migration tasks; complete these tasks in the order shown:
1. "Migrating a message flow that contains WebSphere MQ Everyplace nodes" on page 67
2. "Designing MQe connections" on page 68
3. "Deploying an MQe message flow" on page 70
4. Configuring after MQe message flow deployment (this topic)

Using message flows that contain MQeInput and MQeOutput nodes in Version 6.0
is deprecated. The behavior that is described here is intended only for when you
are deploying from Version 6.0 to a previous version, and to provide a route for
migration. Redesign your flows to remove the MQe nodes and replace them with
MQ nodes that are configured to your own specifications and coordinated with
your MQe gateway configuration.

When you deploy a message flow that contains MQeInput and MQeOutput nodes
on WebSphere Message Broker Version 6.0 the nodes are converted to MQInput
and MQOutput nodes (as described in "Deploying an MQe message flow" on page
70).

When the MQe nodes are converted into MQ nodes at deployment time, the node
properties Queue Manager Name and Queue Name are retained, and you must
create a configuration on the broker queue manager to handle these nodes.

For example, suppose you have a message flow like this simplified one that
contains MQeInput and MQeOutput nodes:
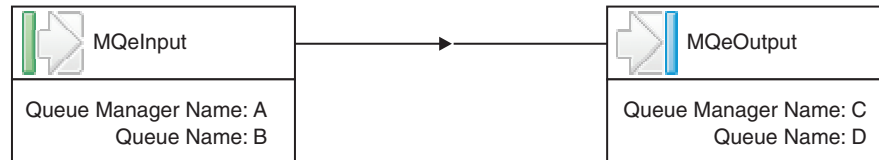
The following diagram shows this message flow connected directly to MQe
devices, as it would be used on a Version 5.0 broker:
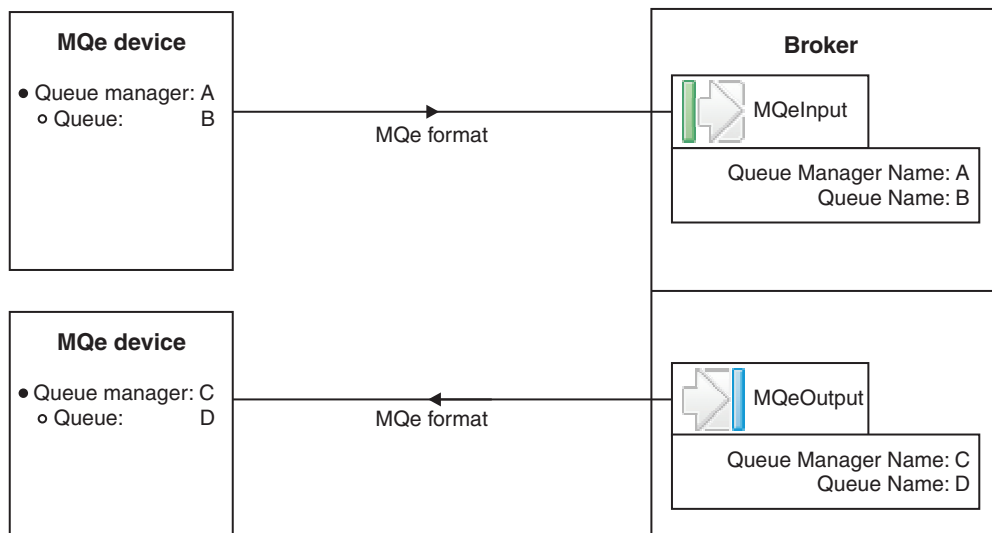
When you deploy this message flow to a Version 6.0 broker, create a configuration
as shown below:

MQe device

- Queue manager: A
  - Queue: B

MQe format →

MQe gateway

(Bridge queue manager)

Bridge queue

MQ format →

Broker

MQInput

Queue Manager Name: A
Queue Name: B

broker-queue-manager
- Queue: B

MQe device

- Queue manager: C
  - Queue: D

← MQe format

MQOutput

Queue Manager Name: C
Queue Name: D

broker-queue-manager
- Remote queue: R
  - Remote Queue Manager: C
  - Remote Queue: *none*
  - Transmission Queue: T
- Transmission queue: T

Listener

← MQ format

Configuration details:

1. Configure your broker queue manager with the following parameters:
   - A local queue named B.
   - A remote queue definition R with these attributes:
     – Remote Queue Manager = C
     – Remote Queue = *none* (leave blank)
     – Transmission Queue = T

   This remote queue definition matches all messages that are destined for any queue on queue manager C and puts them to the transmission queue.

2. Configure your MQe gateway with the following parameters:
   - Provision for accepting messages from an MQe device with queue manager A and queue B, transforming the format from MQe to MQ, and putting them onto a bridge queue.
   - A bridge queue that puts messages to queue B on the broker queue manager.
   - A listener that reads messages from transmission queue T on the broker queue manager.
   - Provision for transforming the message format of the messages on queue T from MQ to MQe format, and sending them to the MQe device with queue manager C and queue D.

For help with configuring WebSphere MQ Everyplace, see the documentation supplied with that product.

Equivalent z/OS customization variables in mqsicompcif and BIPEDIT

If you are migrating from Version 2.1 or Version 5.0 to Version 6.0 on z/OS, customize the Version 6.0 BIPEDIT file to match the Version 2.1 or Version 5.0 mqsicompcif file.

The broker and User Name Server components are set up using the profile BIPEDIT. The following table lists the customization variables that are used in the Version 2.1 and Version 5.0 mqsicompcif file, and the equivalent variables in the Version 6.0 BIPEDIT file so that you can set up the same values.

You can also look at the sample BIPEDIT file.

Version 2.1 and Version 5.0 mqsicompcif customization variables	Version 6.0 BIPEDIT customization variables
DB2_SAMPLE_PGMNAME	++DB2SAMPLEPROGRAM++
DB2_SAMPLE_PGMPLAN	++DB2SAMPLEPROGRAMPLAN++
DB2_DSNACLI_PLANNAME	++DB2DSNACLIPLAN++
DB2_SUBSYSTEM	++DB2SUBSYSTEM++
DB2_LOCATION	++DB2LOCATION++
DB2_TABLE_OWNER	++DB2TABLEOWNER++ ++DB2CURRENTSQLID++ (These two values can have different values)
DB2_STORAGE_GROUP	++DB2STORAGEGROUP++
DB2_STOR_GROUP_VCAT	++DB2LOCATION++
DB2_STOR_GROUP_VOL	Default value in BIPCRDB
DB2_DATABASE	++DB2DATABASE++
DB2_BUFFER_POOL	++DB2BUFFERPOOL++ ++DB2INDEXBP++ ++DB2LOBBP++ (These three values can have different values)
COMPONENT_PDS	++COMPONENTDATASET++
COMPONENT_KEY	++STARTEDTASKNAME++
STPN	++STARTEDTASKNAME++
USER_PROCLIB_PDS	N/A
STEPLIB_NAME	Defined in each JCL
JAVAHOME	++JAVA++
CLASSPATH	Default value in profile
PATH	Default value in profile
LIBPATH	Default value in profile
LILPATH	Default value in profile
DISTHUB_PATH	Default value in profile
LANG	++LOCALE++
LC_ALL	++LOCALE++
TZ	++TIMEZONE++
NLSPATH	Default value in profile
CONSOLE_NLSPATH	Default value in profile

Version 2.1 and Version 5.0 mqsicompcif customization variables	Version 6.0 BIPEDIT customization variables
MC_MESSAGES	Default value in profile
DB2_CONVERSION	++DB2CONVERSION++
USE_ARM	Default value in profile
ARM_ELEMENTNAME	Default value in profile
ARM_ELEMENTTYPE	Default value in profile
_CEE_RUNOPTS	N/A
NNSY_ROOT	Default value in profile
NNSY_CATALOGUES	Default value in profile
NN_CONFIG_FILE_PATH	NN_CONFIG_FILE_PATH in BIPBRPOF

//BIPXXXXX JOB 5655-G97,
// MSGLEVEL=(1,1),
// MSGCLASS=A,
// NOTIFY=&SYSUID
	Defined in each JCL

mqsicreate(broker/usernameserver) command	++INSTALL++
mqsicreate(broker/usernameserver) command	++COMPONENTDIRECTORY++
mqsicreate(broker/usernameserver) command	++COMPONENTNAME++
N/A	++HOME++
N/A	++OPTIONS++
STEPLIB	++WMQHLQ++
mqsicreate(broker/usernameserver) command	++QUEUEMANAGER++
N/A	++COMPONENTPROFILE++
N/A	++XMLTOOLKIT++
STEPLIB	++DB2HLQ++
STEPLIB	++DB2RUNLIB++
N/A	++MQPATH++

Assignments configuration data in an export file

Find the assignments configuration data when you export all your resources from a Version 2.1 Control Center workspace.

This topic describes the assignments configuration data in the XML file that is generated when you export resources by clicking **File** → **Export All in Workspace**. Specifically, the topic tells you how to find the following configuration data for each broker in the broker domain:

- The name of the broker
- The name of each execution group within the broker
- For each execution group within the broker, the name of each message flow that is assigned to the execution group
- For each message flow assigned to an execution group, the following properties:
  - Additional instances

- Commit count
- Commit interval
- Coordinated transaction

The following XML shows the start of an example export file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE XMI SYSTEM "mqsi.dtd">
<XMI xmi.version="1.0">
  <XMI.header>
    <XMI.documentation>
      <XMI.owner>name</XMI.owner>
Broker/a3916f02-f500-0000-0080-8818935dcec8
ExecutionGroup/23bd6f02-f500-0000-0080-8818935dcec8
MessageProcessingNodeType/24c55cb6-ea00-0000-0080-c5b38dd899ad
MessageProcessingNodeType/7e8b6bb6-ea00-0000-0080-c5b38dd899ad
MessageProcessingNodeType/c0656ab6-ea00-0000-0080-c5b38dd899ad
MessageProcessingNodeType/575960b6-ea00-0000-0080-c5b38dd899ad
TopicRoot/TopicRoot
PubSubTopology/PubSubTopology
MessageProcessingNodeType/2f11692e-e900-0000-0080-c5b38dd899ad
MessageProcessingNodeType/8322692e-e900-0000-0080-c5b38dd899ad
MessageProcessingNodeType/4e2f692e-e900-0000-0080-c5b38dd899ad
MessageProcessingNodeType/ff3c692e-e900-0000-0080-c5b38dd899ad
MessageProcessingNodeType/fba95b0e-ef00-0000-0080-aed9bbfe32dc
MessageProcessingNodeType/fc149e14-ef00-0000-0080-aed9bbfe32dc
MessageProcessingNodeType/857e100f-ef00-0000-0080-aed9bbfe32dc
    </XMI.documentation>
  </XMI.header>
  <XMI.content>
    <Broker icon="images/Broker.gif" creator="" version="" collectionPath=""
        versionTimestamp="" longDescription="" status="" versionCreator=""
        creationTimestamp="" shortDescription=""
        xmi.uuid="a3916f02-f500-0000-0080-8818935dcec8"
        xmi.id="a3916f02-f500-0000-0080-8818935dcec8" xmi.label="fred_tp">
      <Broker_queueManager value="ANAME" encoded="false"
          xmi.label="queueManager" type="String" valueMandatory="true"/>
          <ExecutionGroupRef icon="images/ExecutionGroup.gif" xml:link="simple"
          xmi.label="ExecutionGroup" type="ExecutionGroup"
          refType="ExecutionGroup"
          href="ExecutionGroup/23bd6f02-f500-0000-0080-8818935dcec8"
          title="default"/>
    </Broker>
    <ExecutionGroup icon="images/ExecutionGroup.gif" creator="" version=""
        collectionPath="" versionTimestamp="" longDescription="" status=""
        versionCreator="" creationTimestamp="" shortDescription=""
        xmi.uuid="23bd6f02-f500-0000-0080-8818935dcec8"
        xmi.id="23bd6f02-f500-0000-0080-8818935dcec8" xmi.label="default">
      <AssignedMessageFlow icon="images/MessageProcessingNodeType.gif" status=""
          xmi.label="mqsistrmrmswiftflow">
      <AssignedMessageFlow_additionalInstances value="3" encoded="false"
          xmi.label="additionalInstances" type="Integer"
          valueMandatory="false"/>
      <AssignedMessageFlow_commitCount value="5" encoded="false"
          xmi.label="commitCount" type="Integer" valueMandatory="false"/>
      <AssignedMessageFlow_commitInterval value="1" encoded="false"
          xmi.label="commitInterval" type="Integer" valueMandatory="false"/>
      <AssignedMessageFlow_coordinatedTransaction value="no" encoded="false"
          xmi.label="coordinatedTransaction" type="yes_no"
          valueMandatory="false"/>
      <MessageProcessingNodeTypeRef icon="images/MessageProcessingNodeType.gif"
          xml:link="simple" xmi.label="MessageProcessingNodeType"
          type="MessageProcessingNodeType" refType="MessageProcessingNodeType"
          href="MessageProcessingNodeType/24c55cb6-ea00-0000-0080-c5b38dd899ad"
          title="mqsistrmrmswiftflow"/>
    </AssignedMessageFlow>
```

```
      </ExecutionGroup>
      ...
    </XMI.content>
</XMI>
```

The following sections describe how to find the assignments configuration data in an export file.

The name of the broker

For each broker in the broker domain, look for a Broker element within the XMI.content element. The name of the broker is the value of the Broker element's `xmi.label` attribute.

In the example, only one broker is defined. The name of the broker is fred_tp.

The name of each execution group within the broker

For each execution group within a broker, look for an ExecutionGroupRef element within the Broker element. The name of the execution group is the value of the ExecutionGroupRef element's `title`.

In the example, the broker fred_tp has only one execution group, which is the default execution group.

For each execution group in the broker domain, look for an ExecutionGroup element within the XMI.content element. You can determine the broker to which an execution group belongs by examining the value of the ExecutionGroup element's `xmi.uuid` attribute. This value forms part of the value of the `href` attribute on the Broker element's corresponding ExecutionGroupRef element.

In the example, only one execution group in the broker domain is defined, the default execution group for the broker fred_tp. The value of the `xmi.uuid` attribute on the ExecutionGroup element is 23bd6f02-f500-0000-0080-8818935dcec8, and this forms part of the value of the ExecutionGroupRef element's `href` attribute for the broker fred_tp.

The name of each message flow that is assigned to each execution group

For each message flow that is assigned to an execution group, look for an AssignedMessageFlow element within the ExecutionGroup element. The name of the message flow is the value of the AssignedMessageFlow element's `xmi.label` attribute. In the example, only one message flow is assigned to the default execution group for the broker fred_tp. The name of the message flow is mqsistrmrmswiftflow.

The properties of each message flow

For each property of an assigned message flow, look for a corresponding element within the AssignedMessageFlow element. The following table lists each property and its corresponding tag.

Property of an assigned message flow	Corresponding tag
Additional instances	<AssignedMessageFlow_additionalInstances>
Commit count	<AssignedMessageFlow_commitCount>
Commit interval	<AssignedMessageFlow_commitInterval>
Coordinated transaction	<AssignedMessageFlow_coordinatedTransaction>

The value of a property is the value of the corresponding element's `value` attribute.

For example, the following table displays the properties of the message flow mqsistrmrmswiftflow that is assigned to the default execution group within broker fred_tp.

Property	Value
Additional instances	3
Commit count	5
Commit interval	1
Coordinated transaction	no

Migration glossary

This glossary lists differences in terminology between WebSphere Event Broker Version 6.0 and previous versions of the product.

For a full list of terms and abbreviations used by the product documentation, and for definitions that are more complete, see the Glossary of terms and abbreviations.

A

assigned message flow
> In Version 2.1, a message flow that is assigned to an execution group. In Version 6.0, when you add a message flow to a broker archive (bar) file and deploy the bar file to an execution group, the message flow is assigned implicitly to the execution group.

assigned message flow properties
> In Version 2.1, the properties of an assigned message flow. In Version 6.0, you specify the equivalent properties in the deployment descriptor of a broker archive (bar) file.

Assignments view
> In Version 2.1, the user interface in the Control Center that is used to assign message flows to an execution group. In Version 6.0, the equivalent function is provided by the Broker Administration perspective of the workbench. When you add message flows to a broker archive (bar) file and deploy the bar file to an execution group, the message flows are assigned implicitly to the execution group.

B

Broker Administration perspective
> In Version 6.0, a broker administration console that communicates with one or more Configuration Managers. Administrators work in this perspective

to manage the resources (also referred to as domain objects) in the broker
domain that are defined on one or more Configuration Managers. In
Version 2.1, the equivalent function is provided by the Assignments view.

built-in node

In Version 6.0, a message processing node that is supplied by the product.
In Version 2.1, this is referred to as an IBM Primitive node.

C

check in

In Version 2.1, a Control Center action that stores a new or updated
resource in the Configuration Manager. In Version 6.0, the equivalent
check-in function might be provided by an external repository that is
associated with the workbench.

check out

In Version 2.1, a Control Center action that extracts a resource from the
Configuration Manager and locks it for local modification by a user. In
Version 6.0, the equivalent check-out function might be provided by an
external repository that is associated with the workbench.

Configuration Manager

In Version 6.0, the Configuration Manager does not store message flow
definitions. Instead, these definitions are stored in the local file system or
in an external repository that is associated with the workbench. In addition
in Version 6.0 the Configuration Manager no longer uses an external
database to store the domain configuration, but uses an internal repository
instead.

connection

In Version 6.0, this represents the flow of control and data between two
message processing nodes. In Version 2.1, this is referred to as a connector.

connector

In Version 2.1, this represents the flow of control and data between two
message processing nodes. In Version 6.0, this is referred to as a
connection.

Control Center

In Version 2.1, the graphical user interface that provides the function to
define, configure, deploy, and monitor resources. In Version 6.0, this
function is provided, with enhancements, by the Message Brokers Toolkit,
which is also known as the workbench.

I

IBM Primitive node

In Version 2.1, a message processing node that is supplied by the product.
In Version 6.0, this is referred to as a built-in node.

Input Terminal node

In Version 2.1, an IBM Primitive node through which a message is received
by a subflow. In Version 6.0, this function is provided by the Input node.

M

Message Brokers Toolkit

In Version 6.0, the development and operational management environment
for a broker domain. It is also referred to as the workbench. In Version 2.1,
the Control Center provides this function.

O

Operations view
> In Version 2.1, the Control Center view that is used to manage and monitor the brokers in a broker domain. In Version 6.0, the equivalent function is provided by the Domains and Alerts views in the Broker Administration perspective of the workbench.

Output Terminal node
> In Version 2.1, an IBM Primitive node through which a message is propagated by a subflow. In Version 6.0, this function is provided by the Output node.

S

shared data
> In Version 2.1, configuration repository and message repository data that is shared by Control Center users. In Version 6.0, the equivalent functionality might be provided by an external repository that is associated with the workbench.

T

traffic light status icon
> In Version 2.1, an icon next to a message flow, execution group, or broker in the Domain Topology pane of the Operations view in the Control Center. A green traffic light indicates that the resource is running, and a red traffic light indicates that it is not running. In Version 6.0, an alert appears in the Alerts view in the Broker Administration perspective of the workbench only when the resource is not running.

U

unlock
> In Version 2.1, the action to end the checked-out state of a resource in such a way that any changes made to the resource since it was checked out are not stored in the Configuration Manager. In Version 6.0, the equivalent function might be provided by an external repository that is associated with the workbench.

V

view In Version 2.1, a user interface in the Control Center that enables a user to perform a specified task. There are eight views: Message Sets, Message Flows, Topology, Assignments, Topics, Operations, Subscriptions, and Log. In Version 6.0, a view is a component of a perspective in the workbench. To perform a specified task, the user selects the appropriate perspective and then uses a combination of one or more views and an editor.

W

workbench
> See Message Brokers Toolkit.

Part 3. Appendixes

81

Appendix. Notices for WebSphere Event Broker

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032,
Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks in the WebSphere Event Broker information center

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

AIX	CICS	DB2
DB2 Connect	DB2 Universal Database	developerWorks
Domino		Everyplace
FFST	First Failure Support Technology	IBM
IBMLink	IMS	IMS/ESA
Informix	iSeries	i5/OS
Language Environment	Lotus	MQSeries
MVS	NetView	OS/400
OS/390	Passport Advantage	POWER
pSeries	RACF	Rational
Redbooks	RETAIN	RS/6000
SupportPac	System i	S/390
Tivoli	VisualAge	WebSphere
xSeries	z/OS	zSeries

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Pentium are trademarks or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Index

B

backup
   resources
      Version 2.1 11
      Version 5.0 23
BIPEDIT, equivalent variables in
  mqsicompcif 74
broker network
   WebSphere MQ, migrating 55
broker networks
   migration, keeping separate 46
brokers
   migrating
      Version 5.0 30
      WebSphere MQ, migrating 52

C

coexistence 3
commands
   issuing after migration 16
Configuration Manager
   migrating
      Version 5.0 25

G

glossary
   migration 78

H

heterogeneous networks
   creating 46
   deleting brokers in 52

L

leaf nodes
   adding 47

M

Message Brokers Toolkit
   migrating
      Version 5.0 23
message flows
   migration notes 64
migration 3
   brokers
      Version 5.0 30
   command environment, setting
    up 17
   conditions
      Message Brokers Toolkit 7
   configuration data in an export
    file 75

migration *(continued)*
   Configuration Manager
      Version 5.0 25
   glossary 78
   issuing commands after migration 16
   Message Brokers Toolkit
      Version 5.0 23
   message flows
      notes 64
   MQe flows 67
   ODBC
      changing connection
       definitions 33
   planning
      publish/subscribe 45
      Version 2.1 10
      Version 5.0 18
   publish/subscribe applications 45
   supported migration paths 63
   tooling
      Version 5.0 23
   User Name Server
      Version 5.0 37
   user-defined nodes 66
   Version 2.1 7
   Version 5.0 17
   Version 5.0 domain
      high availability 21
      large 20
      multiple 21
      small 20
   WebSphere MQ
      broker network 55
      brokers 52
   WebSphere MQ Event Broker Version
    2.1 12
      preparation 13
   XA resource manager definitions,
    changing 33
MQe connections, designing 68
MQe flows
   configuring after deployment 71
   deploying 70
   migrating 67
mqsicompcif, equivalent variables in
  BIPEDIT 74

O

ODBC
   connection
      changing 33

P

parent nodes
   adding 49
planning
   migration
      publish/subscribe 45

planning *(continued)*
   migration *(continued)*
      Version 2.1 10
      Version 5.0 18
publish/subscribe
   applications
      migrating 45

R

restoring to previous versions 40

T

tooling
   migrating
      Version 5.0 23
trademarks 85

U

upgrading
   supported upgrade paths 63
User Name Server
   migrating
      Version 5.0 37
user-defined nodes
   migrating 66

W

WebSphere MQ
   broker network, migrating 55
   brokers, migrating 52

IBM ®

Printed in USA