WebSphere Message Broker

IBM

Migration

Version 6 Release 1

WebSphere Message Broker

IBM

Migration

Version 6 Release 1

Contents

About this topic collection

This PDF has been created from the WebSphere Message Broker Version 6.1 (November 2008) information center topics. Always refer to the WebSphere Message Broker online information center to access the most current information. The information center is periodically updated on the document update site and this PDF and others that you can download from that Web site might not contain the most current information.

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

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

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

Part 1. Migrating

Migrating and upgrading

To migrate a broker domain to WebSphere® Message Broker Version 6.1 or WebSphere Message Broker with Rules and Formatter Extension Version 6.1, 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 Message Broker support Web page with the latest details available. Look under the heading ″Solve a problem″ for the document ″Problems and solutions when migrating″.

You can migrate to WebSphere Message Broker Version 6.1 or WebSphere Message Broker with Rules and Formatter Extension Version 6.1 from the following products:
- WebSphere Business Integration Event Broker Version 5.0 (Fix Pack 4 or later)
- WebSphere Business Integration Message Broker Version 5.0 (Fix Pack 4 or later)
- WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 (Fix Pack 4 or later)
- WebSphere Event Broker Version 6.0
- WebSphere Message Broker Version 6.0
- WebSphere Message Broker with Rules and Formatter Extension Version 6.0

The instructions apply to all operating systems supported by the Version 5.0 and Version 6.0 products.

The tasks in this section refer to installing Version 6.1; see the Installation Guide for details of how to complete this task.

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

If you have WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 or WebSphere Message Broker with Rules and Formatter Extension Version 6.0, you are not required to complete any additional steps to migrate the extension.

If you are already using WebSphere MQ Publish/Subscribe, migrate your applications to use the publish/subscribe functions provided by WebSphere Message Broker Version 6.1 following the instructions in this section.

This section contains the following topics:
- "Coexistence with previous versions and other products" on page 4
- "Migrating from Version 5.0 products" on page 35
- "Migrating from Version 6.0 products" on page 9
- "Post-migration tasks" on page 61
- "Restoring migrated components to previous versions" on page 65
    WebSphere Message Broker support Web page

3

Coexistence with previous versions and other products

WebSphere Message Broker Version 6.1 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 6.1 components with components from previous versions" on page 5

If you have configured your broker domain such that two or more brokers share a database schema, you must migrate all those brokers to the new version or product at the same time. Brokers that you have created at different versions, or for different products, cannot share a database schema.

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

WebSphere Message Broker Version 6.1 can coexist with a Version 5.0 product, a Version 6.0 product, or both, on the same computer.
- You cannot install more than one instance of a Version 5.0 product on a single computer.
- Component names must be unique on the computer, regardless of version.

When you migrate from a Version 5.0 or Version 6.0 product to WebSphere Message Broker Version 6.1, you do not have to uninstall the Version 5.0 or Version 6.0 product before installing Version 6.1. You can install WebSphere Message Broker Version 6.1 in a different location on the same computer, migrate your components and resources to Version 6.1, and uninstall the Version 5.0 or Version 6.0 product 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 are available.

The following table shows which Version 5.0 and Version 6.0 components can exist on the same computer as Version 6.1 components.

Can the components exist together on the same computer?	Version 6.1 Message Broker Toolkit	Version 6.1 Configuration Manager	Version 6.1 broker	Version 6.1 User Name Server
Version 5.0, Version 5.1, or Version 6.0 Message Broker Toolkit	Yes	Yes	Yes	Yes
Version 5.0 or Version 6.0 Configuration Manager	Yes	Yes	Yes	Yes
Version 5.0 or Version 6.0 broker	Yes	Yes	Yes	Yes

Can the components exist together on the same computer?	Version 6.1 Message Broker Toolkit	Version 6.1 Configuration Manager	Version 6.1 broker	Version 6.1 User Name Server
Version 5.0 or Version 6.0 User Name Server	Yes	Yes	Yes	Yes[1]

Restriction:

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

Coexistence of Version 6.1 components with components from previous versions

With some restrictions, all Version 5.0 and Version 6.0 components can participate in a Version 6.1 broker domain, and all Version 6.1 components can participate in a Version 5.0 or Version 6.0 broker domain.

Because Version 6.1 components can coexist with components from previous versions, you do not have to migrate all of your components at the same time. You can migrate those components in stages that are convenient to you.

The User Name Server has not changed between versions, therefore you can choose not to migrate this component. You can use a previous version of this component with components that you have migrated to Version 6.1.

The following table shows which Version 5.0 and Version 6.0 components can operate with Version 6.1 components, and shows the restrictions that apply.

Can the components operate together in a domain?	Version 6.1 Message Broker Toolkit	Version 6.1 Configuration Manager	Version 6.1 broker
Version 5.0, Version 5.1, or Version 6.0 Message Broker Toolkit	Yes[1]	Yes[2]	Yes[3,4]
Version 5.0 or Version 6.0 Configuration Manager	Yes[2]	(Not applicable)	Yes
Version 5.0 or Version 6.0 broker	Yes[5]	Yes	Yes

Restrictions:

1. If applicable, you must migrate Version 5.0 maps (.mfmap files) and data definitions (.xmi files) using the mqsimigratemfmaps command before you can access them in the Version 6.1 Message Broker Toolkit.

   You can access all other resources that you have created using the Version 5.0, Version 5.1, or Version 6.0 Message Broker Toolkit without migration, but their use might be restricted. For details of

restrictions that apply, see "Conditions for using migrated resources with previous versions of the Message Broker Toolkit."

2. You cannot use a domain user ID to connect a Version 5.0 or Version 5.1 Message Broker Toolkit to a Version 6.1 Configuration Manager, nor to connect a Version 6.1 Message Broker Toolkit to a Version 5.0 Configuration Manager.

3. You can use message flow debugging only between like versions (Version 6.1 toolkit and Version 6.1 broker). You cannot debug message flows on a Version 6.1 broker from a Version 6.0 Message Broker Toolkit. However, you can deploy all resources from a Version 6.0 Message Broker Toolkit to a Version 6.1 broker.

4. When you have deployed to a Version 6.1 broker from a Message Broker Toolkit at any version, you cannot later restore the broker to its previous version.

5. When you deploy from a Version 6.1 Message Broker Toolkit to a Version 5.0 or Version 6.0 broker, you can deploy only unchanged bar files that you have created in the equivalent version of toolkit. For more details about the deployment restrictions that apply, see "Conditions for using migrated resources with previous versions of the Message Broker Toolkit."

Because of these restrictions, you must retain a Version 5.0, Version 5.1, or Version 6.0 Message Broker Toolkit to maintain brokers that are at Version 5.0 or Version 6.0.

All operations controlled through the CMP continue to work with both Version 5.0 and Version 6.0 brokers, including starting, stopping, deleting, and monitoring of deployed resources.

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

WebSphere Message Broker Version 6.1 can interoperate with components of previous versions. Conditions apply when you use migrated development and deployed resources with previous versions of the Message Broker Toolkit.

Version 6.0

If you migrate from WebSphere Message Broker Version 6.0 (or other supported Version 6.0 products), automatic resource migration is performed for data definitions when you import or rebuild a message flow project. See "Migrating a flow containing data definitions" on page 82 for further information.

You do not have to perform any explicit migration tasks on any resources.

Version 5.0

If you migrate from WebSphere Business Integration Message Broker Version 5.0 (or other supported Version 5.0 products), migrate the following resources:

- User-defined extensions: follow the instructions in "Migrating a user-defined node" on page 79.
- Mapping files: follow the instructions in "Migrating message mappings from Version 5.0" on page 75.

You do not have to migrate any other development or deployment resources, such as message flow files, message set definition files, ESQL files, XML Schema files, and broker archive files.

Restrictions after migration

After you have migrated workbench resources to Version 6.1, your use of these resources in the Version 6.0 and Version 5.0 or Version 5.1 Message Broker Toolkit is restricted.

Development resources

Resources in Version 5.0, 5.1, 6.0, and 6.1 formats can coexist in the Version 6.1 workspace.

Resources remain in their original format until you save them in the Version 6.1 Message Broker Toolkit. After saving, all resources are stored in the Version 6.1 format.

You can no longer use these saved resources in earlier versions of the Message Broker Toolkit. Therefore you cannot share development resources between the Version 6.1 and previous versions of the Message Broker Toolkit.

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

Deployment resources

You can deploy BAR files that you edit or create in a Version 6.1 Message Broker Toolkit only to Version 6.1 brokers.

You can deploy BAR files that you created in a Version 5.0 or Version 6.0 to a broker at the same version only if you have not modified the BAR file in the Version 6.1 Message Broker Toolkit. If you change only the deployment descriptor or any included XSLT files, the BAR file version is not modified, and you can continue to deploy the BAR file to a broker at a previous version.

You can import resources that you created in previous versions into your Version 6.1 workspace, but when you include these resources in a BAR file, you can deploy that BAR file only to Version 6.1 brokers.

The following table summarizes which of the files that you can include in a BAR file are deployable by Configuration Manager and accepted by the broker at each version. The Configuration Manager ignores file types that are not supported. A broker rejects broker archive files that contain file types that are supported only by later brokers. For example, a Version 5.0 broker rejects a BAR file that contains a .xml file.

File type	V5.0	V6.0	V6.1
Compiled message flows (.cmf)	Yes	Yes	Yes
Message dictionaries (.dictionary)	Yes	Yes	Yes
XSL style sheets (.xsl, .xslt, .xml)	No	Yes	Yes
Java™ archives (.jar)	No	Yes	Yes
WebSphere TX maps (.mar)	No	No	Yes
XSD archives (.xsdzip)	No	No	Yes
Inbound adapter configuration (.inadapter)	No	No	Yes
Outbound adapter configuration (.outadapter)	No	No	Yes

The Version 6.1 Message Broker Toolkit can coexist with the Version 6.0, Version 5.0, or Version 5.1 Message Broker Toolkit on the same computer. However, because you cannot share development resources between the Version 6.1 Message Broker Toolkit and previous versions of the Message Broker Toolkit, you might find that it is appropriate to create and maintain a different workspace for the Version 6.1 Message Broker Toolkit.

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

Conditions for a Version 5.0 broker participating in a Version 6.1 broker domain

If you choose to include a Version 5.0 broker in a Version 6.1 broker domain, certain conditions and restrictions apply.

If you preserve a broker at the Version 5.0 level of code during migration to WebSphere Message Broker Version 6.1, the broker can subsequently participate in the Version 6.1 broker domain and can be managed by the Version 6.1 Configuration Manager. The Version 5.0 broker can be from either of the following products:

- WebSphere Business Integration Message Broker Version 5.0 (Fix Pack 6 or later)
- WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 (Fix Pack 6 or later)

You cannot preserve a WebSphere MQ Event Broker broker at the Version 5.0 level of code when you migrate a WebSphere Business Integration Event Broker Version 5.0 broker domain to a Version 6.1 broker domain.

The following conditions apply:

- The Version 5.0 broker must use a unique set of database tables. Therefore, if a Version 5.0 broker and a Version 6.1 broker share a database, they must use different database schemas.
- You can use a Version 6.1 Message Broker Toolkit and a Version 5.0 Configuration Manager to deploy to a broker that is part of a WebSphere Business Integration Message Broker Version 5.0 installation or higher. However, the installation of WebSphere Business Integration Message Broker, of which the Version 5.0 Configuration Manager is a part, must be at Fix Pack 5 or higher.

After migration, a Version 5.0 broker can continue to use the same WebSphere MQ queue manager, with the same WebSphere MQ configuration, and with the same release and service level of the WebSphere MQ product code, as it did before migration.

You might need to perform migration tasks for user-defined extensions, mapping files, and data definitions; see "Message flow migration notes" on page 74 for details. You can continue to use other resources, such as message flow files, message set definition files, ESQL files, Java files, XML Schema files, and broker archive files, without change.

When you have migrated Version 5.0 mapping files (.mfmap) to Version 6.1 mapping files (.msgmap) using the mqsimigratemfmaps command, edit them only with a Version 6.1 Message Broker Toolkit.

Migrating from Version 6.0 products

Migrate to WebSphere Message Broker Version 6.1 or WebSphere Message Broker with Rules and Formatter Extension Version 6.1.

You can migrate to WebSphere Message BrokerVersion 6.1 products from the following products:
* WebSphere Event Broker Version 6.0
* WebSphere Message Broker Version 6.0
* WebSphere Message Broker with Rules and Formatter Extension Version 6.0

The instructions in this section apply to all operating systems that are supported by Version 6.0.

If you have installed SupportPac™ IA9Q you must uninstall it before you install WebSphere Message Broker Version 6.1; see "Migrating from SupportPac IA9Q" on page 17.
* "Planning and pre-migration tasks"
* "Migrating from WebSphere Event Broker Version 6.0 to WebSphere Message Broker Version 6.1" on page 18
* "Migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1" on page 18
* "Migrating from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.1" on page 34

Planning and pre-migration tasks

Plan your migration strategy, and complete pre-migration tasks.

The following topics describe what to do before you migrate from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1:
* "Planning for migration from Version 6.0"
* "Backing up WebSphere Message Broker Version 6.0 resources" on page 16
* "Migrating from SupportPac IA9Q" on page 17

Planning for migration from Version 6.0

Plan the order and extent of the migration of components and resources to Version 6.1.

This section explains how to plan your migration. Complete the following steps:
1. Decide how you want to migrate the WebSphere Message Broker Version 6.0 product components:
   a. Find out what is new in Version 6.1 and learn about new and changed function. These changes might affect how you want to use your migrated components in the future.
   b. Check the requirements for other products on which Version 6.1 components might have dependencies. You can find details of prerequisite and other optional products in both the Installation Guide and in the Installation reference section.
   c. If your installation includes a component on a 64-bit platform, or you plan to move one or more components to a 64-bit platform, read the advice provided in "Considerations for 64-bit platforms" on page 11.

d. Decide where you will migrate the product components. Migrate them 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.

The following topics describe different migration scenarios:
- "Planning to migrate a small domain from Version 6.0" on page 13
- "Planning to migrate a large domain from Version 6.0" on page 14
- "Planning to migrate multiple domains from Version 6.0" on page 14
- "Planning to migrate a high-availability domain from Version 6.0" on page 15

e. Decide when to migrate the product components. You might want to preserve some components at the Version 6.0 level for now, and migrate them later. Each migration task includes steps for installing WebSphere Message Broker Version 6.1, so you might decide to carry out the tasks in parallel, rather than completing one task before starting the next task.

f. Decide the order in which you will 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.

A typical order might be:
1) Install a Version 6.1 toolkit.
2) Create a Version 6.1 broker, or migrate an existing broker to Version 6.1.
3) Migrate the Configuration Manager.
4) Migrate further brokers and toolkits.

"Coexistence with previous versions and other products" on page 4 shows you how WebSphere Message Broker Version 6.1 can coexist on the same computer with previous versions of the product, and how Version 6.1 components can operate with components from previous versions.

2. Decide how you will use your existing resources with WebSphere Message Broker Version 6.1.

You do not need to perform any specific tasks to migrate your development and deployment resources, such as message flow files, message set definition files, ESQL files, XML Schema files, and broker archive files. You can start using these resources with WebSphere Message Broker Version 6.1 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 74 and "Message set migration notes" on page 93.

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 6.0 level that you backed up before you started the migration, and any post-migration changes to these resources would be lost. If you migrate your development and test domains before you migrate your production domain, you can identify potential problems and develop a strategy for dealing with further problems.

Each new release can include changes to address product defects that affect external behavior. If your applications depend on undocumented or erroneous behavior, you must test those applications and allow time during your migration schedule to make any necessary changes.

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 6.0 components to ensure that they can be migrated. The pre-migration check identifies potential problems so that you can correct them before you continue 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 confirm to an external standard. If your resources depend on undocumented or incorrect behavior (for example, the ESQL code in a Compute node), 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.

Considerations for 64-bit platforms:

Migration might be affected by 32-bit and 64-bit support on your chosen platforms.

WebSphere Message Broker provides support for both 32-bit and 64-bit platforms. The level of support is different from that provided for previous versions, therefore you might have to take special action during migration.

WebSphere Message Broker supports three types of platforms:
- Platforms that have only 32-bit support. These platforms have 32-bit commands and a 32-bit broker core, and support 32-bit execution groups.
- Platforms that have 32-bit and 64-bit support. These platforms have 64-bit commands and a 64-bit broker core, and support both 32-bit and 64-bit execution groups.
- Platforms that have only 64-bit support. These platforms have 64-bit commands and a 64-bit broker core, and support only 64-bit execution groups.

Read the guidance in the following sections to check if your migration might be affected:
- "Migrating from 32-bit to 64-bit platforms"
- "Migrating 64-bit platforms" on page 12
- "Migrating user defined extensions" on page 13
- "Migrating user exits" on page 13

In the examples shown here, replace *DB2InstanceDir* with the location in which you have installed your DB2® instance.

Migrating from 32-bit to 64-bit platforms:

The steps in this section apply to DB2, Informix®, and Oracle databases, although most of the examples illustrate the changes needed for DB2 databases. If you are migrating a broker from a 32-bit platform to a 64-bit platform:

1. Do not export 32-bit libraries in the library search path (LIBPATH, LD_LIBRARY_PATH, SHLIB_PATH).

2. Remove any environment variable settings that are similar to the following example:
   ```
   export LD_LIBRARY_PATH=DB2InstanceDir/sqllib/lib32:$LD_LIBRARY_PATH
   export SHLIB_PATH=$ORACLE_HOME/lib32:$SHLIB_PATH
   ```

These settings are required only on Linux® on x86 where the required variables are already set in your library search path because they are set by the database profile script.

3. If you had previously set the environment variable MQSI_LIBPATH64 to use 64-bit execution groups with DB2 using the following command, you can now remove this variable.

```
unset MQSI_LIBPATH64
```

4. If you plan to use JMS from 32-bit execution groups in a default 64-bit host environment, you must set the MQ JMS Java library paths on the MQSI_LIBPATH32. For example:

```
export MQSI_LIBPATH32=$MQSI_LIBPATH32:/usr/mqm/lib:/usr/mqm/java/lib
```

5. If you plan to use 32-bit execution groups with a 64-bit database instance, set one of the following variables:

For DB2 Version 8:

```
export MQSI_LIBPATH32=$MQSI_LIBPATH32:DB2InstanceDir/sqllib/lib
```

For DB2 Version 9:

```
export MQSI_LIBPATH32=$MQSI_LIBPATH32:DB2InstanceDir/sqllib/lib32
```

or

```
export MQSI_LIBPATH32=$MQSI_LIBPATH32:$ORACLE_HOME/lib32
```

This path is added as a prefix to the library search path by the 32-bit execution group that is running a 64-bit platform.

If you are migrating, or moving to, only 64-bit executions groups, you do not need to set this variable.

6. The sample ODBC files provided by WebSphere Message Broker have changed; odbc32.ini is provided for 32-bit support and odbc64.ini is provided for 64-bit support. However, default environment variables are no longer set for them in mqsiprofile. You must now set these in your own profile file. Previously, the environment variables were set to:

```
ODBCINI=/var/mqsi/.odbc.ini
ODBCINI64=/var/mqsi/odbc64.ini
```

Change these to:

```
ODBCINI=<Fully qualified path to your copy of the 64-bit odbc64.ini file>
ODBCINI32=<Fully qualified path to your copy of the 32-bit odbc32.ini file>
```

ODBCINI64 can be removed because it is no longer required.

Migrating 64-bit platforms:

WebSphere Message Broker Version 6.1 supports only 64-bit mode on some platforms. This section describes additional steps that you must take when you migrate brokers on or to these platforms.

Linux on POWER™ and Solaris on x86-64
DB2 must run in 64-bit mode on these platforms. If you used WebSphere Message Broker Version 6.0 with DB2 Version 8 in 32-bit mode, and the database instance resides on the same computer as the broker, you must migrate the local DB2 instance and databases to 64-bit before you migrate the broker.

If you are using DB2 Version 8 or Version 9 in 64-bit mode, you do not have to complete any additional tasks.

When you have completed these tasks, continue to migrate your brokers using the instructions provided.

Linux on System z®

1. You cannot migrate brokers that you have created on a computer running a 31-bit Linux on System z operating system to Version 6.1. If you have a Version 6.0 broker in that environment, you must create a broker in a 64-bit installation and deploy your resources to the new broker.

2. If you have created a Version 6.0 broker on a computer running a 64-bit Linux on System z operating system, follow the previous instructions for Linux on POWER and Solaris on x86-64.

AIX®, HP-UX, and Solaris on SPARC - 64-bit installations only

If you are using DB2 Version 8.2, and you have a 32-bit execution group on a Version 6.0 broker on AIX, HP-UX, or Solaris on SPARC which accesses a local 32-bit DB2 instance, you must migrate the local DB2 instance to a 64-bit instance before you migrate the broker.

If you are using DB2 Version 9 on your Version 6.0 systems, the database supports only 64-bit instances, and you do not have to take any further action.

Migrating user defined extensions:

If you use any C user-defined extensions, and you are not running with a 32-bit execution group, recompile these programs to be 64-bit compliant.

Use the mqsichangebroker command, with the appropriate flag, to specify the directory in which you have stored the user-defined extension files:
- For 32-bit extensions specify the -l flag.
- For 64-bit extensions specify the -r flag.

Migrating user exits:

If you have created C language user exits, and you are not running with a 32-bit execution group, recompile these programs to be 64-bit compliant.

Use the mqsichangebroker command, with the appropriate flag, to specify the directory in which you have stored the user exit files:
- For 32-bit exits specify the -x flag.
- For 64-bit exits specify the -b flag.

Planning to migrate a small domain from Version 6.0:

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

A small domain typically 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 Message Broker Version 6.1 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 6.0 components can coexist with Version 6.1 components within a single domain. See "Coexistence with previous versions and other products" on page 4 for more information.

Migrate components in the order that best suits your environment. For example, you might typically migrate the three major components in this order:

1. Instances of the Message Broker Toolkit
2. Configuration Manager
3. Brokers

If you migrate instances of the Message Broker Toolkit first, you can begin to develop message flows that use features that are new in Version 6.1. If you migrate the Configuration Manager, administrators can prepare your Version 6.1 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.

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

Planning to migrate a large domain from Version 6.0:

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 Broker Toolkit and a test Configuration Manager
2. Migrate a test broker
3. Migrate more instances of the Message Broker Toolkit
4. Migrate more brokers
5. Migrate the remaining components

The migration order is flexible because WebSphere Message Broker Version 6.1 components can coexist with Version 6.0 components in a single domain. See "Coexistence with previous versions and other products" on page 4 for more information.

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

Planning to migrate multiple domains from Version 6.0:

Migrate multiple domains that perform different functions in your environment.

A typical configuration for WebSphere Message 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 do not handle 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 considered 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.

Migrate development and test domains 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 use new functions, you must migrate the test and production domains before you deploy the new message flows to them.

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

Message Broker Toolkit Version 6.1 can administer Version 6.0 domains; similarly, Message Broker Toolkit Version 6.0 can administer Version 6.1 domains

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

Planning to migrate a high-availability domain from Version 6.0:

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 Message Broker Version 6.1. One solution is to install and configure WebSphere Message Broker Version 6.1 while the previous version of the product is running in parallel, and then switch to WebSphere Message Broker Version 6.1. Install WebSphere Message Broker Version 6.1 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 these guidelines 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 6.0" on page 14):

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 Message Broker Version 6.1 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 6.1, create a Version 6.1 Configuration Manager and broker on as many computers as are required to mirror the production domain.

2. Move one functional area to Version 6.1 by migrating the Message Broker 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, and 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 6.1. 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 Broker Toolkit Version 6.1. You can then administer the new Version 6.1 components. Refer to the product-specific migration topics for instructions.

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

Backing up WebSphere Message Broker Version 6.0 resources

Back up your resources before you start migrating components to Version 6.1.

Before you start:

Read the following topics to determine your migration strategy:

- "Planning to migrate a small domain from Version 6.0" on page 13
- "Planning to migrate a large domain from Version 6.0" on page 14
- "Planning to migrate multiple domains from Version 6.0" on page 14
- "Planning to migrate a high-availability domain from Version 6.0" on page 15

Before you carry out any migration tasks, back up your WebSphere Message Broker Version 6.0 resources by completing the following steps:

1. Back up the configuration repository. Use the mqsibackupconfigmgr command.
2. Back up the broker database tables. Use the documented procedures that are provided by your database supplier.
3. Back up your Message Broker Toolkit workspace and resources; for example, message flow files, message set definition files, Java files, ESQL files, mapping files, XML Schema 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 Message Broker Version 6.0 information center.

After you have backed up your WebSphere Message Broker Version 6.0 resources, proceed to one of the following topics:
• "Migrating from WebSphere Event Broker Version 6.0 to WebSphere Message Broker Version 6.1" on page 18
• "Migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1" on page 18
• "Migrating from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.1" on page 34

Migrating from SupportPac IA9Q

If you installed SupportPac IA9Q, that introduced the SRRetrieveITService and SRRetrieveEntity nodes, this SupportPac must be removed before you can upgrade to WebSphere Message Broker Version 6.1.

You must remove SupportPac IA9Q or you might encounter problems when you deploy, or use, the EndpointLookup and RegistryLookup nodes that replace the SRRetrieveITService and SRRetrieveEntity nodes.

Before you install WebSphere Message Broker Version 6.1 ensure that you complete the following steps:
• Write down the node properties within your SRRetrieveITService and SRRetrieveEntity nodes so that you can use them when you create the EndpointLookup and RegistryLookup nodes in WebSphere Message Broker Version 6.1.
• Remove flows containing the SRRetrieveITService and SRRetrieveEntity nodes from your WebSphere Message Broker Version 6.0.
• Remove the dynamic Cache Notification component (the Cache Synchronization WebSphere Message Broker BAR file), if you deployed it.
• Edit the broker's classpath, in bin/mqsiprofile.cmd, to remove the following files:
   – sibc.jms.jar
   – sibc.jndi.jar
   – ibm-jaxrpc-client.jar
   – Solaris   sibc.orb.jar (on the Solaris platform)

When you have edited the broker's classpath delete the files from your system.

- Remove the ServiceRegistryMessageBrokerNodes_1.0.0.par file.
- Remove the following files from the shared classes (<MQSI working path>/shared-classes) directory:
  - wsrr.properties
  - sdo-int.jar
  - ServiceRegistryClient.jar
  - WMBc4WSRR_1.0.0.jar
- Remove the Message Broker Toolkit plugin files from the Toolkit. This directory was created when you unzipped SupportPac IA9Q, for example, C:\IBM\MessageBrokersToolkit\6.0\evtoolkit\eclipse\plugins\ com.ibm.sr.mb.nodes_1.0.1

You can now return to "Planning and pre-migration tasks" on page 9.

Migrating from WebSphere Event Broker Version 6.0 to WebSphere Message Broker Version 6.1

Migrating from WebSphere Event Broker Version 6.0 follows the same process as migrating from WebSphere Message Broker Version 6.0.

To migrate from WebSphere Event Broker Version 6.0 to WebSphere Message Broker Version 6.1, follow the instructions in "Migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1," replacing all references to "WebSphere Message Broker Version 6.0" with "WebSphere Event Broker Version 6.0".

In some places, the instructions for migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1 give information about the file types to which the instructions apply. Some of the file types in WebSphere Message Broker do not exist in WebSphere Event Broker; ignore any file types that are specific to WebSphere Message Broker.

For example, if the instructions ask you to perform an action on message flow files, message set definition files, ESQL files, mapping files, XML Schema files, and broker archive files, you need to perform the action on message flow files and broker archive files only, because the other files types apply to WebSphere Message Broker only.

Migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1

Complete the tasks to migrate from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1.

Before you start
- Read about coexistence with previous versions and other products
- Plan your migration
- Back up your Version 6.0 resources

Your installation of WebSphere Message Broker Version 6.0 does not have to be at a particular service level; Version 6.0 (GA) and all fix packs are supported.

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 need to install WebSphere Message Broker Version 6.1.

To migrate a single component, complete the task for that component. To migrate all components, complete all of the tasks listed,

- Migrating the Message Broker Toolkit
- Migrating a Configuration Manager
- Migrating a broker
- Migrating a User Name Server
- Migrating a DatabaseInstanceMgr

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

Migrating the Message Broker Toolkit from Version 6.0 to Version 6.1

Follow the instructions to migrate the Message Broker Toolkit Version 6.0.

Although you cannot migrate the Message Broker Toolkit component from WebSphere Message Broker Version 6.0 to Version 6.1, you can install Version 6.1 to coexist with Version 6.0, and migrate the workspace and the resources that it contains.

To migrate the Message Broker Toolkit workspace from Version 6.0 to Version 6.1, complete the following steps:

1. Install WebSphere Message Broker Version 6.1 in a different location from WebSphere Message Broker Version 6.0.
2. When you start the Message Broker Toolkit Version 6.1 for the first time, you are prompted to enter a workspace location. Enter the directory that contains theMessage Broker ToolkitVersion 6.0 workspace that you want to migrate.
   - If your message flows include user-defined nodes, see "Migrating a user-defined node" on page 79.
   - If your message flow projects contain data definitions for ESQL and mappings, see "Migrating a flow containing data definitions" on page 82.

   The workspace and resources are now available in the Version 6.1 Message Broker Toolkit.

The resources in the workspace retain their Version 6.0 content until you edit or rebuild them. You can deploy resources to Version 6.0 brokers without making any changes. You can also deploy these resources to Version 6.1 brokers.

If you open and save these resources, they are automatically updated to Version 6.1 format, and you can deploy them only to brokers that have been migrated to Version 6.1.

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 Broker Toolkit Version 6.1. To resolve these problems, configure access control lists (ACLs) for users who are running the Message Broker Toolkit Version 6.1. Follow the instructions in Considering security for the workbench.

When you have migrated the Message Broker Toolkit, ensure that you have migrated the rest of the components:

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

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 Configuration Manager from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1

Follow the instructions to migrate the Configuration Manager.

To migrate a Configuration Manager from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1, see the appropriate topic for your operating system:

- Distributed systems
- z/OS®

Migrating a Version 6.0 Configuration Manager to Version 6.1 on distributed systems:

Migrate a Version 6.0 Configuration Manager to Version 6.1.

You can migrate the Configuration Manager to the new version on the same computer. If you want to move the Configuration Manager to another computer (with the same or a different operating system), you can backup the Version 6.0 Configuration Manager and restore it in a Version 6.1 installation.

Select the topic that is appropriate to your environment:

- "Migrating the Configuration Manager on the same computer"
- "Moving the Configuration Manager to a second computer" on page 21

Migrating the Configuration Manager on the same computer:

Before you start

If the Configuration Manager 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 a WebSphere Message BrokerVersion 6.0 broker on distributed operating systems to Version 6.1, complete the following steps:

1. Install WebSphere Message Broker Version 6.1 as a new instance on the computer where Version 6.0 is installed.
2. In your Version 6.0 command environment, stop the Version 6.0 Configuration Manager by using the mqsistop command.
3. Set up the correct Version 6.1 command environment:
   - `Linux` `UNIX` On Linux and UNIX® systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

- $\boxed{\text{Windows}}$ On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

4. Run the mqsimigratecomponents command to migrate your Version 6.0 Configuration Manager. For example:

   ```
   mqsimigratecomponents ConfigMgr
   ```

5. Run the mqsistart command to start the Version 6.1 Configuration Manager.

Moving the Configuration Manager to a second computer:

Before you start

To complete this task you must have either a Version 6.0 or a Version 6.1 Message Broker Toolkit available.

If the Configuration Manager 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 a Version 6.0 Configuration Manager on a distributed operating system to Version 6.1 to a different computer, complete the following steps:

1. Stop the Version 6.0 Configuration Manager by using the mqsistop command.
2. Backup the Configuration Manager repository. Use the mqsibackupconfigmgr command to save the component's internal resource and setup information in an archive file (the -a parameter).
3. Install WebSphere Message Broker Version 6.1 on the target computer.
4. Copy the archive file created by the mqsibackupconfigmgr command from the source computer to the target computer.
5. If the target computer is a distributed system:
   a. Set up the correct Version 6.1 command environment:

      - $\boxed{\text{Linux}}$ $\boxed{\text{UNIX}}$ On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

      - $\boxed{\text{Windows}}$ On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

   b. Create the Configuration Manager by using the mqsicreateconfigmgr command.
   c. Restore the repository to the Configuration Manager on the second computer. Use the mqsirestoreconfigmgr command to retrieve the saved repository from the archive file.
6. If the target computer is a z/OS system:
   a. Create the Configuration Manager by customizing and submitting the BIPCRCM job.
   b. Restore the repository to the Configuration Manager on the second computer. Customize and submit the BIPRSCM job to retrieve the saved repository from the archive file.
   c. Copy the file service.properties file from the source computer to the target computer. Store the service.properties file in the appropriate directory, defined in the mqsirestoreconfigmgr command description.
7. Start the Version 6.1 Configuration Manager by using the mqsistart command.
8. In the workbench, start a deploy action to all brokers. This action causes the Version 6.1 Configuration Manager to subscribe to broker status messages.

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 Broker 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, your Configuration Manager has been migrated successfully. If they are unsuccessful, resolve any problems before you migrate further components or make configuration changes.

When you have migrated or moved the Configuration Manager, ensure that you have migrated the rest of the components:

- Migrate the Message Broker Toolkit
- Migrate the broker
- Migrate the User Name Server
- Migrate the Windows DatabaseInstanceMgr

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 6.0 Configuration Manager to Version 6.1 on z/OS:

Migrate a Version 6.0 Configuration Manager to Version 6.1.

Before you start
- Ensure that you are familiar with the steps involved in creating a Configuration Manager on z/OS.
- Ensure that you are familiar with the mqsimigratecomponents command which the JCL (Job Control Language) file uses the to migrate a Configuration Manager.

To migrate a WebSphere Message Broker Version 6.0 Configuration Manager to Version 6.1 on z/OS:

1. Install WebSphere Message Broker Version 6.1 as a new instance on the computer where Version 6.0 is installed.
2. Stop the Version 6.0 Configuration Manager.
3. Create a new Configuration Manager PDSE.
4. Copy all Configuration Manager JCL from the Version 6.1 installed SBIPPROC and SBIPSAMP PDSEs to the new Configuration Manager PDSE and customize them all.

   You must take a backup copy of your Version 6.0 ENVFILE, which is stored in the directory referenced by the *++HOME++* JCL variable, before you submit BIPGEN from the Version 6.1 component data set. Subsequently, if you want to migrate back to Version 6.0, restore ENVFILE, either from the backup file

location, or by submitting BIPGEN from the Version 6.0 component data set, before you start the Version 6.0 Configuration Manager.

   a. Customize the BIPEDIT file using values that are defined in the Version 6.0 BIPEDIT file.

   b. If you have made any other changes to the environment file ENVFILE, copy your changes to BIPCPROF in the component data set. Submit the BIPGEN job to create the environment file, ENVFILE.

   c. Customize and submit the BIPMGCMP (mqsimigratecomponents command) job.

5. Copy the started task JCL (BIPCMGRP) 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. Start the Version 6.1 Configuration Manager. The verification program runs when you start the component.

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

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

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

Migrating a broker from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1

Follow the instructions to migrate the broker.

To migrate a broker from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1, see the appropriate topic for your operating system:

- Distributed systems
- z/OS

You do not have to redeploy resources to a broker that you have migrated. The broker starts existing message flows as soon as you start it.

Migrating a Version 6.0 broker to Version 6.1 on distributed operating systems:

Migrate a broker to use the enhanced facilities available in a Version 6.1 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 6.0 broker's application logic onto a separate Version 6.1 broker.

Select the topic that is appropriate to your environment:
- "Migrating a broker on the same computer"
- "Migrating a broker's application logic to another installation" on page 24

Migrating a broker on the same computer:

Before you start

Before you migrate a broker, check that you do not have any aggregations in progress. When you migrate a broker to WebSphere Message Broker Version 6.1, all live data that is being stored for aggregations in progress is lost.

If the broker 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 a Version 6.0 broker on a distributed operating system to Version 6.1, complete the following steps:

1. Install WebSphere Message Broker Version 6.1 on the same computer as Version 6.0.
2. In your Version 6.0 command environment, stop the Version 6.0 broker by using the mqsistop command.
3. Stop all channels that are connected to the Version 6.0 broker.
4. Follow the instructions in "Updating your ODBC definitions when migrating" on page 26 to update the ODBC definitions file to Version 6.1 format.
5. Set up the correct Version 6.1 command environment:

   - **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - **Windows** On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

6. Enter the mqsimigratecomponents command to migrate the broker. For example:

   ```
   mqsimigratecomponents Broker1
   ```

7. If you are running Version 6.1 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:

     ```
     export LD_LIBRARY_PATH=[DB2instanceDir]/sqllib/lib32:$LD_LIBRARY_PATH
     export SHLIB_PATH=$ORACLE_HOME/lib32:$SHLIB_PATH
     ```

   - If you set the environment variable MQSI_LIBPATH64 on your Version 6.0 operating system to use 64-bit execution groups, you can now remove this variable. For example:

     ```
     export MQSI_LIBPATH64=[DB2InstanceDir]/sqllib/lib64:$MQSI_LIBPATH64
     ```

   - If you want to continue to use 32-bit execution groups, and you use DB2 for your broker database, you must set an environment variable.

     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 6.1 broker by using the mqsistart command.

Migrating a broker's application logic to another installation:

Before you start

To complete this task you must have either a Version 6.0 or a Version 6.1 Message Broker Toolkit available.

Before you migrate a broker, check that you do not have any aggregations in progress. When you migrate a broker to Version 6.1, all live data that is being stored for aggregations in progress is lost.

If the broker 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 a Version 6.0 broker on a distributed operating system to Version 6.1 at a different location on the same computer, or to a different computer, complete the following steps:

1. Install WebSphere Message Broker Version 6.1 either as a new instance on the computer where Version 6.0 is installed, or on a different computer.
2. Update the ODBC definitions file to Version 6.1 format by following the instructions in "Updating your ODBC definitions when migrating" on page 26.
3. Set up the correct Version 6.1 command environment:

   - Linux   UNIX On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - Windows On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

4. Create a Version 6.1 broker by using the mqsicreatebroker command; give it a name that is different from the name of the Version 6.0 broker.

   Broker names must be unique within a domain. Brokers cannot share queue managers. You cannot migrate a broker to a queue manager that is already being used by an existing broker.
5. Start the Version 6.1 broker by using the mqsistart command.
6. 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
7. Write a list of the execution groups that you have on the Version 6.0 broker, and create these same execution groups on the new broker. Use either the Version 6.0 or Version 6.1 Message Broker Toolkit to complete this step.
8. Deploy the message flows and message sets of the Version 6.0 broker to the Version 6.1 broker. Use either the Version 6.0 or Version 6.1 Message Broker Toolkit to complete this step.
9. Configure all other relevant properties of the Version 6.0 broker on the Version 6.1 broker. For example, you might need to configure properties that you set by using the mqsichangeproperties command, or as a result of using certain nodes, such as Publication or SCADA nodes.
10. Stop the Version 6.0 broker. If you have installed Version 6.1 on the same computer, you must enter the mqsistop command in the correct Version 6.0 command environment.
11. Remove the Version 6.0 broker from the workbench.
12. Redeploy the topology.
13. Delete the Version 6.0 broker by using the mqsideletebroker command.

When you have migrated the broker, ensure that you have migrated the rest of the components:

- Migrate the Message Broker Toolkit
- Migrate the Configuration Manager
- Migrate the User Name Server
- Migrate the Windows DatabaseInstanceMgr

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 and user 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:

- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 5.0 to Version 6.1" on page 54
- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 6.0 to Version 6.1.0.2 (or later)" on page 27

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 6.1, 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 6.1 command environment.

In your new Version 6.1 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: 64-bit considerations.

Sample ODBC files are provided; .odbc32.ini is supplied for 32-bit connections and odbc64.ini is supplied for 64-bit connections. However, the default environment variables have changed. The following values are set in mqsiprofile for Version 5.0 and Version 6.0:

```
ODBCINI=/var/mqsi/.odbc.ini
ODBCINI64=/var/mqsi/odbc64.ini
```

These environment variables are no longer set in mqsiprofile. Add these environment variables to your own user profile:

```
ODBCINI32=<fully qualified path to your copy of the 32-bit odbc32.ini file>
ODBCINI=<fully qualified path to your copy of the 64-bit odbc64.ini file>
```

If you are creating or changing 64-bit ODBC definitions for Oracle, use the Version 6.1 sample ODBC files to ensure that you set the correct parameters. The 64-bit connections require different parameters to those required in previous releases.

If you are using XA to enable coordinated transactions, also update your qm.ini file for WebSphere MQ:

- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 5.0 to Version 6.1" on page 54
- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 6.0 to Version 6.1.0.2 (or later)"

You can select a specific version of the broker by placing all user-written scripts in the directory containing the /common/profiles subdirectory, and by adding conditional logic using the variables:

- MQSI_VERSION
- MQSI_VERSION_V for the version
- MQSI_VERSION_R for the release
- MQSI_VERSION_M for the modification
- MQSI_VERSION_F for the fix level

For example, on AIX, the scripts must be placed in the directory /var/mqsi/common/profiles. The following script, on AIX, identifies which version of the broker is called. You can also use logic instead of the echo statements:

```
#!/usr/bin/ksh
# This script should be put into /var/mqsi/common/profiles
# it will be invoked by WMB mqsiprofile
# Test which WMB version called us
  if [[ ${MQSI_VERSION_V} = "6" && ${MQSI_VERSION_R} = "0" ]];then
    echo "WMB user profile script invoked by WMB Version 6.0"
  fi
  if [[ ${MQSI_VERSION_V} = "6" && ${MQSI_VERSION_R} = "1" ]];then
    echo "WMB user profile script invoked by WMB Version 6.1"
  fi
```

If you revert to a previous version of WebSphere Message 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 67. If you are reverting to Version 6.0, follow the instructions in "Restoring components and resources to Version 6.0" on page 65.

Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 6.0 to Version 6.1.0.2 (or later):

Update definitions for Oracle and Sybase databases when you migrate brokers from Version 6.0 to Version 6.1.0.2 or later.

WebSphere Message Broker Version 6.1 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 supplied. You must make a number of changes as a result of the later versions of the ODBC drivers for Oracle and Sybase databases.

- If you are migrating a broker from Version 6.0 to Version 6.1 on a Windows system, update the ODBC definition for each broker and the user data source name as described below. Complete this task for Oracle and Sybase databases only.

- If you are migrating a broker from Version 6.0 to Version 6.1 on Linux or UNIX systems, create a new ODBC configuration file that contains definitions for each of your existing data sources. Use the Version 6.1 template to complete this task, as described in Connecting to a database from Linux and UNIX systems: 64-bit considerations or Connecting to a database from Linux and UNIX systems: 32-bit considerations. Do not change the existing definition file.

- If you have a message flow that updates an Oracle or Sybase database within a global unit of work that is coordinated by the broker queue manager, you must also change the XA resource manager definition for the database.

- If you are reverting a broker from Version 6.1 back to Version 6.0, you must reverse these changes.

Make the required changes before you run the Version 6.1 mqsimigratecomponents command. If you are reverting to Version 6.0, keep the new definitions until after you have finished running the mqsimigratecomponents command. The following sections describe the changes in detail.

Changing the ODBC connection definitions:

Linux and UNIX systems:

Create a new ODBC definitions file by following the instructions in Connecting to a database from Linux and UNIX systems: 64-bit considerations or Connecting to a database from Linux and UNIX systems: 32-bit considerations. Before you run the Version 6.1 commands, check that your ODBCINI environment variable points to the new file and not to the existing file.

Windows systems:

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 5.30 32-BIT Oracle
Sybase	MQSeries DataDirect Technologies 5.30 32-BIT Sybase Wire Protocol

Changing the XA resource manager definitions:

Linux and UNIX systems:

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/UKor8dtc20.so`
	
	`or`
	`SwitchFile=UKor8tdc20.so`
		`SwitchFile=UKor8dtc23.so`

Sybase	`SwitchFile=`
	`    /usr/opt/mqsi/merant/lib/UKasedtc20.so`
	
	`or`
	`SwitchFile=UKasedtc20.so`
		`SwitchFile=UKasedtc23.so`

`Solaris` `Linux` Complete the following changes on Solaris and Linux.

DBMS	Change ...	To ...

Oracle	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKor8dtc20.so`
	
	`or`
	`SwitchFile=UKoradtc20.so`
		`SwitchFile=UKor8dtc23.so`

Sybase	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKasedtc20.so`
	
	`or`
	`SwitchFile=UKasedtc20.so`
		`SwitchFile=UKasedtc23.so`

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

DBMS	Change ...	To ...

Oracle	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKor8dtc20.so`
	
	`or`
	`SwitchFile=UKor8tdc20.so`
		`SwitchFile=UKor8dtc23.so`

DBMS	Change ...	To ...
Sybase	SwitchFile= /opt/mqsi/merant/lib/UKase8dtc20.so or SwitchFile=UKasedtc20.so	SwitchFile=UKasedtc23.so

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

DBMS	Change ...	To ...
Oracle	SwitchFile= /opt/mqsi/merant/lib/UKor8dtc20.sl	SwitchFile=UKor8dtc20.sl
Sybase	SwitchFile= /opt/mqsi/merant/lib/UKase8dtc20.sl	SwitchFile=UKasedtc20.sl

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 Message Broker.

DBMS	Change ...	To ...
Oracle	WBIMB\bin\ukor8dtc20.dll	WBIMB\bin\ukor8dtc23.dll
Sybase	WBIMB\bin\ukase20.dll	WBIMB\bin\ukase23.dll

Migrating a Version 6.0 broker to Version 6.1 on z/OS:

Migrate a z/OS broker to use the enhanced facilities available in a Version 6.1 broker domain.

Before you start
- Ensure that you are familiar with the steps involved in creating a broker on z/OS.
- Ensure that you are familiar with the mqsimigratecomponents command which the JCL (Job Control Language) file uses the to migrate a broker.
- Ensure that you do not have any aggregations in progress. When you migrate a broker to Version 6.1, any live data that is being stored for aggregations in progress is lost.
- If the broker 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 a WebSphere Message Broker Version 6.0 broker to Version 6.1 on z/OS:

1. Stop the Version 6.0 broker.
2. Back up the broker database tables. Use the documented procedures that DB2 provides.
3. Create a new broker PDSE.
4. Copy all broker JCL from the Version 6.1 installed SBIPPROC and SBIPSAMP PDSEs to the new broker PDSE and customize them all. See Customizing the broker JCL for more information.

   You must take a backup copy of your Version 6.0 ENVFILE, which is stored in the directory referenced by the *++HOME++* JCL variable, before you submit BIPGEN from the Version 6.1 component data set. Subsequently, if you want to migrate back to Version 6.0, restore ENVFILE, either from the backup file location, or by submitting BIPGEN from the Version 6.0 component data set, before you start the Version 6.0 broker.

   a. Customize the BIPEDIT file using values that are defined in the Version 6.0 BIPEDIT file.
   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 (mqsimigratecomponents) job. This job migrates the registry, queue, and broker database. As part of the database migration, database tables are created, deleted, and dropped, therefore you must you have the correct DB2 privileges. These privileges are the same as those needed to run the mqsicreatebroker command.
5. Copy the started task JCL (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 6.1 broker.

When you have migrated the broker, ensure that you have migrated the rest of the components:
- Migrate the Message Broker 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 Message Broker Version 6.0 to WebSphere Message Broker Version 6.1

Follow the instructions to migrate the User Name Server.

To migrate a User Name Server from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1, see the appropriate topic for your operating system:
- Distributed systems
- z/OS

Migrating a Version 6.0 User Name Server to Version 6.1 on distributed systems:

Migrate a Version 6.0 User Name Server to Version 6.1.

The User Name Server component has not been changed for the Version 6.1 product. This component also retains no persistent state. Therefore you do not have to perform migration for this component:

- You can migrate the User Name Server if you choose by following the instructions in this topic.
- You can delete and recreate the User Name Server.
- You can use the Version 5.0 User Name Server with your Version 6.1 broker domain.

If you want to move your User Name Server to another computer, create a new one on that computer and delete the current User Name Server when you have included the new one in your broker domain.

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, complete the following steps:

1. Install WebSphere Message Broker Version 6.1, in a different location on the computer where Version 6.0 is installed.
2. In your Version 6.0 command environment, stop the Version 6.0 User Name Server by using the mqsistop command.
3. Set up the correct Version 6.1 command environment:

   - **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - **Windows** On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.
4. Enter the mqsimigratecomponents command to migrate the User Name Server. For example:

   ```
   mqsimigratecomponents usernameserver
   ```
5. Start the Version 6.1 User Name Server by using the mqsistart command.

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

- Migrate the Message Broker Toolkit
- Migrate the Configuration Manager
- Migrate the broker
- Migrate the DatabaseInstanceMgr

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 6.0 User Name Server to Version 6.1 on z/OS:

Migrate a Version 6.0 User Name Server to Version 6.1.

The User Name Server component has not been changed for the Version 6.1 product. This component also retains no persistent state. Therefore you do not have to perform migration for this component. You can choose to migrate the component, or you can delete and recreate it on the same computer, or on a

different computer. If you create another User Name Server, you can delete the original component when you have included the new one in your broker domain.

Before you start

- Ensure that you are familiar with the steps involved in creating a User Name Server on z/OS.
- Ensure that you are familiar with the mqsimigratecomponents command which the JCL (Job Control Language) file uses the to migrate a User Name Server.

To migrate a WebSphere Message Broker Version 6.0 User Name Server to Version 6.1 on z/OS:

1. Stop the Version 6.0 User Name Server.
2. Create a new User Name Server PDSE.
3. Copy all User Name Server JCL from the Version 6.1 installed SBIPPROC and SBIPSAMP PDSEs to the new User Name Server PDSE and customize them all.

   You must take a backup copy of your Version 6.0 ENVFILE, which is stored in the directory referenced by the *++HOME++* JCL variable, before you submit BIPGEN from the Version 6.1 component data set. Subsequently, if you want to migrate back to Version 6.0, restore ENVFILE, either from the backup file location, or by submitting BIPGEN from the Version 6.0 component data set, before you start the Version 6.0 User Name Server.

   a. Customize the BIPEDIT file using values that are defined 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 6.1 User Name Server.

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

- Migrate the Message Broker Toolkit
- Migrate the Configuration Manager
- Migrate the broker
- Migrate the DatabaseInstanceMgr

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 6.0 DatabaseInstanceMgr to Version 6.1

Migrate the DatabaseInstanceMgr to Version 6.1 to access the upgraded Derby database.

The DatabaseInstanceMgr is available only on Windows to manage Derby and DB2 databases. If you have run the Default Configuration wizard or the mqsicreatedb command, a DatabaseInstanceMgr has been created for you as a Windows service. Migrate this component to the latest level of Derby database support.

Take the following steps:

1. Install WebSphere Message Broker Version 6.1 , either as a new instance on the computer where WebSphere Message Broker Version 6.0 is installed, or on a different computer.

2. Stop all brokers on this computer that have a database that is defined as a Derby database. Stop all brokers that you have created using the Default Configuration wizard, and those that access a Derby database that you have created with the mqsicreatedb command before you stop the DatabaseInstanceMgr.

3. Stop the DatabaseInstanceMgr.

4. Launch a Version 6.1 Command Console and enter the mqsimigratecomponents command to migrate the DatabaseInstanceMgr. For example:

   `mqsimigratecomponents DatabaseInstanceMgr`

5. Start the Version 6.1 DatabaseInstanceMgr.

You can access the updated DatabaseInstanceMgr from both Version 6.0 brokers and Version 6.1 brokers; you can migrate the brokers that use this DatabaseInstanceMgr at a later time.

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

- Migrate the Message Broker Toolkit
- Migrate the Configuration Manager
- 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 from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.1

Migrating from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 follows the same process as migrating from WebSphere Message Broker Version 6.0.

To migrate a broker domain from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.1, follow the instructions in "Migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1" on page 18, replacing all references to "WebSphere Message Broker" with "WebSphere Message Broker with Rules and Formatter Extension".

The Rules and Formatter Extension database has not changed in Version 6.1. The component is updated when you migrate to Version 6.1, but you do not have to perform any additional migration steps.

Migrating from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 to WebSphere Message Broker Version 6.1: If you have not used the Rules and Formatter Extension functions in WebSphere Message Broker with Rules and Formatter Extension Version 6.0, or you do not want to continue using them

after the migration, you can migrate from WebSphere Message Broker with Rules and Formatter Extension Version 6.0 to WebSphere Message Broker Version 6.1:

If you have deployed any message flows that use the Rules and Formatter Extension function, remove these message flows from the affected brokers before you migrate those brokers to WebSphere Message Broker Version 6.1, following the instructions in "Migrating from WebSphere Message Broker Version 6.0 to WebSphere Message Broker Version 6.1" on page 18.

Migrating from Version 5.0 products

Migrate to WebSphere Message Broker Version 6.1 or WebSphere Message Broker with Rules and Formatter Extension Version 6.1 from Version 5.0 products.

You can migrate from the following products:
- WebSphere Business Integration Event Broker Version 5.0
- WebSphere Business Integration Message Broker Version 5.0
- WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0

The instructions apply to all operating systems supported by the Version 5.0 products.

This section contains the following topics:
- "Planning and pre-migration tasks"
- "Migrating from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Message Broker Version 6.1" on page 41
- "Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1" on page 42
- "Migrating from WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.0" on page 61

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 6.1:
- "Planning for migration from Version 5.0"
- "Backing up WebSphere Business Integration Message Broker Version 5.0 resources" on page 40

Planning for migration from Version 5.0
Plan the order and extent of the steps you must complete to migrate components and resources to Version 6.1.

This section explains how to plan your migration:
1. Decide how you want to migrate the product components:
   a. Find out what is new in Version 6.1 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 6.1 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 37
- "Planning to migrate a large domain from Version 5.0" on page 38
- "Planning to migrate multiple domains from Version 5.0" on page 38
- "Planning to migrate a high-availability domain from Version 5.0" on page 39

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 4 shows you how WebSphere Message Broker Version 6.1 can coexist on the same computer with previous versions of the product, and how Version 6.1 components can operate with components from previous versions.

2. Decide how you will use your existing resources with WebSphere Message Broker Version 6.1.

Application resources

> You must perform specific tasks to migrate your development and deployment resources, such as message flow files, message set definition files, ESQL files, XML Schema files, and broker archive files, only if you have user-defined extensions, data definitions, and mapping files,
>
> You can start using these resources with WebSphere Message Broker Version 6.1 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 74 and "Message set migration notes" on page 93.
>
> - You must upgrade all Version 5.0 and Version 5.1 user-defined node projects to work with Message Broker Toolkit Version 6.1. Upgrade a project by cleaning it: click **Project** → **Clean**. When you clean a project, the extension point that is required by Version 6.1 to compile the ESQL files contained in the user-defined extension is created in the project plugin.xml file.
> - Migrate Version 5.0 data definitions using the mqsimigratemfmaps command, or start the toolkit to initiate this migration.
> - Migrate Version 5.0 mapping files (.mfmap) to Version 6.1 mapping files (.msgmap) using the mqsimigratemfmaps command.

After you start using your current resources in the Message Broker Toolkit Version 6.1, reuse of those resources is subject to restrictions in Message Broker Toolkit Version 5.0 or Version 5.1. For more information, see "Conditions for using migrated resources with previous versions of the Message Broker Toolkit" on page 6.

Code page converters

The changes in converters between WebSphere Business Integration Message Broker Version 5.0 and WebSphere Message Broker Version 6.1 are significant, therefore the set of converters from the previous level has been included with WebSphere Message Broker Version 6.1.

If you make use of converters in your Version 5.0 environment, you might need to take additional steps to ensure that you can continue to use them after migration. See Using converters from a previous level of the product for details of this task.

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. If you migrate your development and test domains before you migrate your production domain, you can identify problems like this and develop a strategy for dealing with further problems.

   Each new release can include changes to address product defects that affect external behavior. If your applications depend on undocumented or erroneous behavior, you must test those applications and allow time during your migration schedule to make any necessary changes.

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 (for example, the ESQL code in a Compute node), 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 Message Broker Version 6.1 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 6.1 components within a single domain. See "Coexistence with previous versions and other products" on page 4 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 Broker Toolkit
2. Configuration Manager
3. Brokers

If you migrate instances of the Message Broker Toolkit first, you can begin to develop message flows that use features that are new in Version 6.1. If you migrate the Configuration Manager, administrators can prepare your Version 6.1 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 Broker Toolkit and a test Configuration Manager
2. Migrate a test broker
3. Migrate more instances of the Message Broker Toolkit
4. Migrate more brokers
5. Migrate the remaining components

The migration order is flexible because Version 6.1 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 Broker Toolkit, because some files cannot be read using Message Broker Toolkit Version 5.0 or Version 5.1 after they have been saved using Message Broker Toolkit Version 6.1. See "Coexistence with previous versions and other products" on page 4 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 Message 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 Broker Toolkit can administer multiple domains, be careful when migrating instances of the Message Broker Toolkit not to affect any domains that have not been migrated yet.

Message Broker ToolkitVersion 6.1 can administer Version 5.0 domains; similarly, Message Broker ToolkitVersion 5.0 can administer Version 6.1 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 Message Broker Version 6.1. One solution is to install and configure WebSphere Message Broker Version 6.1 while the previous version of the product is running in parallel, and then switch to WebSphere Message Broker Version 6.1. Install WebSphere Message Broker Version 6.1 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 38):

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 Message Broker Version 6.1 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 6.1, create a Version 6.1 Configuration Manager and broker on as many computers as are required to mirror the production domain.

2. Move one functional area to Version 6.1 by migrating the Message Broker 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 6.1. 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 Broker Toolkit Version 6.1. You can then administer the new Version 6.1 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 Message 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 37
- "Planning to migrate a large domain from Version 5.0" on page 38
- "Planning to migrate multiple domains from Version 5.0" on page 38
- "Planning to migrate a high-availability domain from Version 5.0" on page 39

Before you carry out any migration tasks, back up your WebSphere Business Integration Message 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 Broker Toolkit workspace and resources; for example, message flow files, message set definition files, Java files, ESQL files, mapping files, XML Schema 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 Message Broker Version 5.0 information center.

After you have backed up your WebSphere Business Integration Message Broker Version 5.0 resources, proceed to one of the following topics:
- "Migrating from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Message Broker Version 6.1"
- "Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1" on page 42
- "Migrating from WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.0" on page 61

Migrating from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Message Broker Version 6.1

Migrating from WebSphere Business Integration Event Broker Version 5.0 follows the same process as migrating from WebSphere Business Integration Message Broker Version 5.0.

To migrate from WebSphere Business Integration Event Broker Version 5.0 to WebSphere Message Broker Version 6.1, follow the instructions in "Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1" on page 42, replacing all references to "WebSphere Business Integration Message Broker Version 5.0" with "WebSphere Business Integration Event Broker Version 5.0".

In some places, the instructions for migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1 give information about the file types to which the instructions apply. Some of the file types in WebSphere Message Broker do not exist in WebSphere Event Broker; ignore any file types that are specific to WebSphere Message Broker.

For example, if the instructions ask you to perform an action on message flow files, message set definition files, ESQL files, mapping files, XML Schema files, and broker archive files, you need to perform the action only on message flow files and broker archive files, because the other files types apply only to WebSphere Message Broker.

Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1

Migrate product components from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1.

Before you start

Before you start migration:
- Ensure that your installation of WebSphere Business Integration Message 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 Message Broker Version 6.1.

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 Message Broker Version 6.1, 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 Broker 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.

Migrating the Message Broker Toolkit from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1

Migrate the Message Broker Toolkit Version 5.0.

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

When you start using your resources in Message Broker Toolkit Version 6.1, restrictions apply to the use of these same resources again with Message Broker

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

To migrate Message Broker Toolkit Version 5.0 to Version 6.1:

1. Install WebSphere Message Broker Version 6.1 in a different location from WebSphere Business Integration Message Broker Version 5.0.

2. Migrate Version 5.0 mappings and data definitions using the mqsimigratemfmaps command.

3. When you start the Message Broker Toolkit Version 6.1 for the first time, you are prompted to enter a workspace location. Enter the directory where the Message Broker 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 6.1 format automatically when you save them for the first time. Message Broker Toolkit Version 5.0 cannot read message flows that have been saved in Version 6.1 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 86.

   - If your message flows use XML style sheets, follow the instructions in "Migrating style sheets and XML files from Version 5.0" on page 81.

   - If you have previously activated validation on a message flow, but have reset the Validation property of the input node to None, you must ensure that the Parse Timing property is set to the default value On Demand before you migrate the message flow.

     Although the Parse Timing property is ignored by a Version 5.0 broker if Validation is set to None, it is not ignored in Version 6.1, and the message flow will fail.

     If Validation is set to None and Parse Timing is not set to the default value, first reset Validation to another value. The Parse Timing property is enabled and you can set it back to the default value On Demand, then reset the Validation property back to None.

     Migrate the message flow with these updated values; it is migrated correctly and with valid settings for Version 6.1.

   - Version 5.0 message sets are converted automatically to Version 6.1 format when you save them for the first time. Message Broker Toolkit Version 5.0 cannot read message sets that have been saved in Version 6.1 format.

4. Clean and rebuild your workspace.

5. To migrate user-defined nodes from Version 5.0, import the user-defined node project into the Version 6.1 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 Broker Toolkit.

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 Broker Toolkit Version 6.1. To resolve these problems, configure access control lists for users who are running the Message Broker Toolkit Version 6.1. Follow the instructions in Considering security for the workbench.

When you have migrated the Message Broker 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 61 for information about tasks that you might want to perform after migration.

Migrating a Configuration Manager from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1

Follow the instructions to migrate the Configuration Manager Version 5.0.

In WebSphere Message Broker Version 6.1, 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 Message 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 6.1 on the same computer:

Migrate a Version 5.0 Configuration Manager to Version 6.1 on the same system.

Before you start

Read about "Coexistence with previous versions and other products" on page 4.

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 6.1:

1. Install WebSphere Message Broker Version 6.1 in a location other than where Version 5.0 is installed.

2. In your Version 5.0 command environment, stop the Version 5.0 Configuration Manager by using the mqsistop command.
3. Set up the correct Version 6.1 command environment:

   - `Linux` `UNIX` On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - `Windows` On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

4. Run the mqsimigratecomponents command to migrate your Version 5.0 Configuration Manager. For example:

   ```
   mqsimigratecomponents ConfigMgr
   ```

5. Run the mqsistart command to start the Version 6.1 Configuration Manager.

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 Message Broker Version 6.1 in a location other than where Version 5.0 is installed.
2. In your Version 5.0 command environment, stop the Version 5.0 Configuration Manager by using the mqsistop command.
3. Set up the correct Version 6.1 command environment:

   - `Linux` `UNIX` On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - `Windows` On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

4. Create a new Version 6.1 Configuration Manager by using the mqsicreateconfigmgr command.

   You must specify a different name and a different queue manager name for the Version 6.1 Configuration Manager.

   Use the **-n** parameter of the mqsicreateconfigmgr command to specify the Version 5.0 Configuration Manager database. This option initializes the Version 6.1 Configuration Manager with the domain configuration details from the Version 5.0 Configuration Manager.

5. Configure WebSphere MQ to connect the Version 6.1 Configuration Manager to the broker network. For example, you might need to configure channels and transmission queue definitions.

6. Start the Version 6.1 Configuration Manager by using the mqsistart command.

7. Deploy the complete topology by using either the Message Broker Toolkit or the mqsideploy command. This action subscribes all brokers in the domain with the new Version 6.1 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 6.1 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 Broker Toolkit Version 6.1.

To resolve these problems, configure access control lists for users who are running the Message Broker Toolkit Version 6.1. Follow the instructions in Considering security for the workbench.

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 Broker Toolkit and connect it to the domain that is managed by this 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 6.1 on a different computer that has DB2 installed:

Migrate a Version 5.0 Configuration Manager to Version 6.1 on another system.

Before you start

Read about "Coexistence with previous versions and other products" on page 4.

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 6.1 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 Message Broker Version 6.1 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 6.1 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/db2810/db2810).

3. Stop the Version 5.0 Configuration Manager.

4. Create a Version 6.1 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 6.1 Configuration Manager, domain information from the Version 5.0 Configuration Manager database is copied to the Version 6.1 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 6.1 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 6.1 Configuration Manager.

7. Deploy the complete topology using either the Message Broker Toolkit or the mqsideploy command. This action subscribes all brokers in the domain with the new Version 6.1 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 6.1 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 Broker 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 6.1 on a different computer that does not have DB2 installed:

Migrate a Version 5.0 Configuration Manager to Version 6.1 on another system.

Before you start

Read about "Coexistence with previous versions and other products" on page 4.

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 6.1 on the same computer:
   a. Install WebSphere Message Broker Version 6.0 in a location other than where WebSphere Business Integration Message Broker Version 5.0 is installed.
   b. Stop the Version 5.0 Configuration Manager.
   c. Launch a Version 6.1 Command Console and enter the mqsimigratecomponents command to migrate your Version 5.0 Configuration Manager.
   d. Start the Version 6.1 Configuration Manager.

      When you start the Version 6.1 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 6.1 Configuration Manager. You do not need to do anything, and the DB2 database is not modified.
2. Stop the Version 6.1 Configuration Manager, and make a copy of its internal repository using the mqsibackupconfigmgr command.
3. Install WebSphere Message Broker Version 6.0 on the second computer.
4. On the second computer, create a Version 6.1 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 6.1 Configuration Manager, choose a different queue manager name.
5. On the second computer, configure WebSphere MQ to allow the Version 6.1 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 6.1 Configuration Manager repository with the repository that you backed up from the original computer.
7. On the second computer, start the Version 6.1 Configuration Manager.
8. Deploy the complete topology using either the Message Broker Toolkit or the mqsideploy command. This action subscribes all brokers in the domain with the new Version 6.1 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 6.1 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 Broker 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 Message Broker Version 5.0 to WebSphere Message Broker Version 6.1

Follow the instructions to migrate the broker.

To migrate a broker from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1, see the appropriate topic for your operating system:
- Distributed systems
- z/OS

Migrating a Version 5.0 broker to Version 6.1 on distributed operating systems:

Migrate a broker to use the enhanced facilities available in a Version 6.1 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 6.1 broker.

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

Select the topic that is appropriate to your environment.
- "Migrating a broker on the same computer"
- "Migrating a Version 5.0 broker's application logic to another installation" on page 51

Migrating a broker on the same computer:

Before you start

Before you migrate a broker, ensure that you do not have any aggregations in progress. When you migrate a broker to WebSphere Message Broker Version 6.1, all live data that is being stored for aggregations in progress is 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 6.1 on the same computer, complete the following steps:

1. Install WebSphere Message Broker Version 6.1 in a different location from WebSphere Business Integration Message Broker Version 5.0.
2. In your Version 5.0 command environment, stop the Version 5.0 broker.
3. Stop all channels that are connected to the Version 5.0 broker.
4. Update the ODBC definitions file to Version 6.1 format by following the instructions in "Updating your ODBC definitions when migrating" on page 26.
5. Set up the correct Version 6.1 command environment:

   - **`Linux`** **`UNIX`** On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - **`Windows`** On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

6. Enter the mqsimigratecomponents command to migrate the broker. For example:

   ```
   mqsimigratecomponents Broker1
   ```

7. If you are running Version 6.1 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 6.1 broker by 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 6.1 Message Broker Toolkit available.

Before you migrate a broker, ensure that you do not have any aggregations in progress. When you migrate a broker to Version 6.1, all live data that is being stored for aggregations in progress is 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 6.1 at a different location on the same computer, or to a different computer, complete the following steps:

1. Install WebSphere Message Broker Version 6.1 either as a new instance on the computer where WebSphere Business Integration Message 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 6.1 format by following the instructions in "Updating your ODBC definitions when migrating" on page 26.

3. Set up the correct Version 6.1 command environment:

   - Linux UNIX On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.

   - Windows On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.

4. Create a Version 6.1 broker by using the mqsicreatebroker command; 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.

5. Start the Version 6.1 broker by using the mqsistart command.

6. 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

7. 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 6.1 Message Broker Toolkit to complete this step.

8. Deploy the message flows and message sets of the Version 5.0 broker to the Version 6.1 broker. Use either the Version 5.0 or Version 6.1 Message Broker Toolkit to complete this step.

9. Configure all other relevant properties of the Version 5.0 broker on the Version 6.1 broker. For example, you might need to configure properties that you set by using the mqsichangeproperties command, or as a result of using certain nodes, such as Publication or SCADA nodes.

10. Stop the Version 5.0 broker. If you have installed Version 6.1 on the same computer, you must enter the mqsistop command in the correct Version 5.0 command environment.

11. Remove the Version 5.0 broker from the workbench.

12. Redeploy the topology.

13. Delete the Version 5.0 broker by using the mqsideletebroker command.

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 and user 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:

- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 5.0 to Version 6.1" on page 54
- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 6.0 to Version 6.1.0.2 (or later)" on page 27

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 6.1, 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 6.1 command environment.

In your new Version 6.1 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: 64-bit considerations.

Sample ODBC files are provided; .odbc32.ini is supplied for 32-bit connections and odbc64.ini is supplied for 64-bit connections. However, the default environment variables have changed. The following values are set in mqsiprofile for Version 5.0 and Version 6.0:

```
ODBCINI=/var/mqsi/.odbc.ini
ODBCINI64=/var/mqsi/odbc64.ini
```

These environment variables are no longer set in mqsiprofile. Add these environment variables to your own user profile:

```
ODBCINI32=<fully qualified path to your copy of the 32-bit odbc32.ini file>
ODBCINI=<fully qualified path to your copy of the 64-bit odbc64.ini file>
```

If you are creating or changing 64-bit ODBC definitions for Oracle, use the Version 6.1 sample ODBC files to ensure that you set the correct parameters. The 64-bit connections require different parameters to those required in previous releases.

If you are using XA to enable coordinated transactions, also update your qm.ini file for WebSphere MQ:

- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 5.0 to Version 6.1" on page 54

- "Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 6.0 to Version 6.1.0.2 (or later)" on page 27

You can select a specific version of the broker by placing all user-written scripts in the directory containing the /common/profiles subdirectory, and by adding conditional logic using the variables:

- MQSI_VERSION
- MQSI_VERSION_V for the version
- MQSI_VERSION_R for the release
- MQSI_VERSION_M for the modification
- MQSI_VERSION_F for the fix level

For example, on AIX, the scripts must be placed in the directory /var/mqsi/common/profiles. The following script, on AIX, identifies which version of the broker is called. You can also use logic instead of the echo statements:

```
#!/usr/bin/ksh
# This script should be put into /var/mqsi/common/profiles
# it will be invoked by WMB mqsiprofile
# Test which WMB version called us
  if [[ ${MQSI_VERSION_V} = "6" && ${MQSI_VERSION_R} = "0" ]];then
    echo "WMB user profile script invoked by WMB Version 6.0"
  fi
  if [[ ${MQSI_VERSION_V} = "6" && ${MQSI_VERSION_R} = "1" ]];then
    echo "WMB user profile script invoked by WMB Version 6.1"
  fi
```

If you revert to a previous version of WebSphere Message 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 67. If you are reverting to Version 6.0, follow the instructions in "Restoring components and resources to Version 6.0" on page 65.

Changing the ODBC connection and XA resource manager definitions for a broker migrated from Version 5.0 to Version 6.1:

Update definitions for Oracle and Sybase databases.

WebSphere Message Broker Version 6.1 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 6.1 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 5.0 to Version 6.1 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 ODBC connection definitions on Linux and UNIX systems" on page 55.

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 5.0 to Version 6.1 on a Windows system, update the ODBC definition for each broker and the user data source name as described in "Changing the ODBC connection definitions on Windows systems" on page 57. 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 57.

- If you are restoring a broker from Version 6.1 back to Version 5.0, you must reverse these changes.

Changing the 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: 64-bit considerations. Do not change the existing definition file.

Before you run Version 6.1 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/UKor8dtc18.so```
	
	or
	
	``SwitchFile=
	  /usr/opt/mqsi/merant/lib/UKor8dtc20.so```
	
	or
	
	```SwitchFile=UKor8tdc20.so```
		```SwitchFile=UKor8dtc23.so```

DBMS	Change ...	To ...

Sybase	`SwitchFile=`
	`    /usr/opt/mqsi/merant/lib/UKase18.so`
	
	or
	
	`SwitchFile=`
	`    /usr/opt/mqsi/merant/lib/UKasedtc20.so`
	
	or
	
	`SwitchFile=UKasedtc20.so`
		`SwitchFile=UKasedtc23.so`

Solaris   Linux   Complete the following changes on Solaris and Linux.

DBMS	Change ...	To ...

Oracle	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKor8dtc18.so`
	
	or
	
	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKor8dtc20.so`
	
	or
	
	`SwitchFile=UKor8tdc20.so`
		`SwitchFile=UKor8dtc23.so`

Sybase	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKase18.so`
	
	or
	
	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKasedtc20.so`
	
	or
	
	`SwitchFile=UKasedtc20.so`
		`SwitchFile=UKasedtc23.so`

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

DBMS	Change ...	To ...

Oracle	`SwitchFile=`
	`    /opt/mqsi/merant/lib/libSwitchOracle.sl`
	
	or
	
	`SwitchFile=`
	`    /opt/mqsi/merant/lib/UKor8dtc20.sl`
	
	or
	
	`SwitchFile=UKor8tdc20.sl`
		`SwitchFile=UKor8dtc23.sl`

DBMS	Change ...	To ...

Sybase	`SwitchFile=`
	`  /opt/mqsi/merant/lib/libSwitchSybase.sl`
	
	or
	
	`SwitchFile=`
	`/opt/mqsi/merant/lib/UKasedtc20.sl`
	
	or
	
	`SwitchFile=UKasedtc20.sl`
		`SwitchFile=UKasedtc23.sl`

Changing the 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 5.30 32-BIT Oracle
Sybase	MQSeries DataDirect Technologies 5.30 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 Message Broker.

DBMS	Change ...	To ...
Oracle	`WMBIB\bin\ukor8dtc18.dll or WMBIB\bin\` `ukor8dtc20.dll`	`WBIMB\bin\ukor8dtc23.dll`
Sybase	`WMBIB\bin\ukasedtc18.dll or WMBIB\bin\` `ukasedtc20.dll`	`WBIMB\bin\ukase23.dll`

Migrating a Version 5.0 broker to Version 6.1 on z/OS:

Migrate a broker to use the enhanced facilities available in a Version 6.1 broker domain.

Before you start

- Ensure that you are familiar with the steps involved in creating a broker on z/OS.
- The JCL (Job Control Language) migration job file uses the mqsimigratecomponents command to migrate a broker on z/OS. This command takes many parameters, which you must understand fully before you attempt to migrate the broker.
- Ensure that you do not have any aggregations in progress. When you migrate a broker to Version 6.1, any live data that is being stored for aggregations in progress are 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 WebSphere Message Broker Version 5.0 broker to Version 6.1 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 6.1 installed SBIPPROC and SBIPSAMP PDSEs to the new broker PDSE and customize them all. See Customizing the broker JCL for more information.

   You must take a backup copy of your Version 5.0 ENVFILE, which is stored in the directory referenced by the *++HOME++* JCL variable, before you submit BIPGEN from the Version 6.1 component data set. Subsequently, if you want to migrate back to Version 5.0, restore ENVFILE, either from the backup file location, or by submitting BIPGEN from the Version 5.0 component data set, before you start the Version 5.0 broker.

   a. Customize the BIPEDIT file with values that are defined in the mqsicompcif file for the Version 5.0 broker. See "Equivalent z/OS customization variables in mqsicompcif and BIPEDIT" on page 99 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, queues, 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 6.1 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 Message Broker Version 5.0 to WebSphere Message Broker Version 6.1

Follow the instructions to migrate the User Name Server.

To migrate a User Name Server from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1, see the appropriate topic for your operating system:

- Distributed systems
- z/OS

Migrating a Version 5.0 User Name Server to Version 6.1 on distributed systems:

Migrate a Version 5.0 User Name Server to Version 6.1.

The User Name Server component has not been changed for the Version 6.1 product. This component also retains no persistent state. Therefore you do not have to perform migration for this component:

- You can migrate the User Name Server if you choose by following the instructions in this topic.
- You can delete and recreate the User Name Server.
- You can use the Version 5.0 User Name Server with your Version 6.1 broker domain.

If you want to move your User Name Server to another computer, create a new one on that computer and delete the current User Name Server when you have included the new one in your broker domain.

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 WebSphere Message BrokerVersion 6.1 in a different location on the computer where WebSphere Business Integration Message Broker Version 5.0 is installed.
2. In your Version 5.0 command environment, stop the Version 5.0 User Name Server.
3. Set up the correct Version 6.1 command environment:
   - `Linux` `UNIX` On Linux and UNIX systems, open a new shell and run the environment profile mqsiprofile for this Version 6.1 installation.
   - `Windows` On Windows, click **Start**, and open the Command Console that is associated with this Version 6.1 installation.
4. Enter the mqsimigratecomponents command to migrate the User Name Server. For example:

```
mqsimigratecomponents usernameserver
```

5. Start the Version 6.1 User Name Server by 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 6.1 on z/OS:

Migrate a Version 5.0 User Name Server to Version 6.1.

Before you start

* Ensure that you are familiar with the steps involved in creating a User Name Server on z/OS.
* The JCL job uses the mqsimigratecomponents command to migrate a User Name Server on z/OS. This command takes many parameters, which you must understand fully before you attempt to migrate the User Name Server.

To migrate a WebSphere Message Broker Version 5.0 User Name Server to Version 6.1 on z/OS:

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 6.1 installed SBIPPROC and SBIPSAMP PDSEs to the new User Name Server PDSE and customize them all.

   You must take a backup copy of your Version 5.0 ENVFILE, which is stored in the directory referenced by the *++HOME++* JCL variable, before you submit BIPGEN from the Version 6.1 component data set. Subsequently, if you want to migrate back to Version 5.0, restore ENVFILE, either from the backup file location, or by submitting BIPGEN from the Version 5.0 component data set, before you start the Version 5.0 User Name Server.

   a. Customize the BIPEDIT file using values that are defined in the mqsicompcif file for the Version 5.0 User Name Server. "Equivalent z/OS customization variables in mqsicompcif and BIPEDIT" on page 99 lists the customization variables that are used in the Version 5.0 mqsicompcif file, and the equivalent variables in the Version 6.1 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 6.1 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.

Migrating from WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.0

To migrate a broker domain from WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 to WebSphere Message Broker with Rules and Formatter Extension Version 6.0, follow the instructions in "Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1" on page 42, replacing all references to "WebSphere Message Broker" with "WebSphere Message Broker with Rules and Formatter Extension".

For instructions on how to migrate the Rules and Formatter Extension, see the New Era of Networks documentation.

You must migrate the Rules and Formatter extension immediately after migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.0. Do not carry out any operations between migrating WebSphere Message Broker and migrating the Rules and Formatter Extension.

Migrating from WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 to WebSphere Message Broker Version 6.0

Migrate from WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0 to WebSphere Message Broker Version 6.0 only if you have not used the Rules and Formatter functions in WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0, or you do not want to continue using them after the migration.

If you have deployed any message flows that use the Rules and Formatter function that is available in WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0, remove these message flows from the affected brokers before the brokers are migrated to WebSphere Message Broker Version 6.0.

After you have removed these message flows from the affected brokers, follow the instructions in "Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.1" on page 42.

Post-migration tasks

After you have migrated to Version 6.1, finish setting up your environment.

Test the WebSphere Message Broker Version 6.1 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"
- "Removing obsolete messages from error processing" on page 63
- "Uninstalling SupportPac IA9O" on page 64
- "Setting up a command environment" on page 64

When you have done these tasks, you can delete the Version 5.0 or Version 6.0 components and remove the installed code. Access the Version 5.0 or Version 6.0 information center in the workbench for details of these tasks.

Removing unwanted files and database tables after migration

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

Before you start

Before you start this task, make sure that the migration was successful and that you no longer need to retain the flexibility to use the code and resources of the product from which you migrated.

To remove unwanted files and database tables after migration, complete the following steps:

1. If you no longer need the WebSphere Message Broker Version 6.0 or Version 5.0 product, remove it from the computer. Refer to the relevant information center for instructions about uninstalling the product.
2. If you no longer use prerequisite software products that are required by Version 6.0 or Version 5.0, remove them.

   For example, you might no longer need Rational® Agent Controller (RAC). If you have migrated from Version 5.0, you might choose to remove DB2 from the computer on which the Configuration Manager was installed.

   For instructions about how to remove these or other products, 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 the Version 5.0 Configuration Manager.

   In Version 6.1, the Configuration Manager no longer uses an external database to store the domain configuration but uses an internal repository instead. Therefore you cannot use your Version 5.0 Configuration Manager DB2 database tables with the Version 6.1 Configuration Manager.

   To remove the Version 5.0 Configuration Manager DB2 database tables, drop the following database tables using the DB2 DROP TABLE command. If the Configuration Manager tables are in a DB2 database by themselves, drop the whole database 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

Removing obsolete messages from error processing

Update message processing to remove obsolete messages.

WebSphere Message Broker Version 6.1 components do not generate some of the diagnostic messages (BIP messages) that were generated in previous versions. If you have applications that check for specific messages, for example to automate error reporting or to identify the occurrence of specific error conditions, you must update these routines.

Version 6.1 components do not generate the following messages:

```
BIP1133
BIP2825
BIP3305
BIP5179
BIP5182
BIP5196
BIP5197
BIP5198
BIP5219
BIP5220
BIP5276
BIP5422
BIP5423
BIP5425
BIP5438
BIP5439
BIP5440
BIP5441
BIP5442
BIP5443
BIP5444
BIP5508
BIP5509
BIP5510
BIP5511
BIP5512
BIP5513
BIP5514
BIP5515
BIP8621
BIP8622
BIP8623
```

```
BIP8624
BIP8625
BIP8626
BIP8627
BIP8628
BIP8629
BIP8630
BIP8631
BIP8632
BIP8633
BIP8634
BIP8635
BIP8636
BIP8637
BIP8638
BIP8639
BIP8640
BIP8641
BIP8642
BIP8643
BIP8644
BIP8645
BIP8646
BIP8647
BIP8648
BIP8649
BIP8650
BIP8651
BIP8652
BIP8653
BIP8654
BIP8655
BIP8656
BIP8657
BIP8658
BIP8659
BIP8826
```

Uninstalling SupportPac IA9O

Uninstall the SupportPac after you have installed Version 6.1.

SupportPac IA9O includes the SOAPExtract node and the SOAPEnvelope node, which are included with WebSphere Message Broker Version 6.1. If you have been using this SupportPac, you must uninstall it after you have installed Version 6.1.

On the runtime platform, delete file mqsisoap.par from the directory to which you copied it when you installed the SupportPac. You do not need to make any changes on the system on which you have installed the Message Broker Toolkit.

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 Message Broker Version 6.1 from a previous version. A profile is provided to help you set up the environment.

If appropriate, you can extend the initialization performed by this profile; for example, for broker or user databases, or for other products that you want to use within the broker.

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 Message 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 6.1. By restoring components to the previous level, you can fix the problem with Version 6.1. 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 6.1, 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 6.1 broker, then restore it to a previous level. Nor can you create a Version 5.0 broker, migrate it to Version 6.1, then restore it to Version 6.0.

The topics in this section explain how to restore components and resources that you have migrated from Version 6.0 or Version 5.0 products back to their original state:
- "Restoring components and resources to Version 6.0"
- "Restoring components and resources to Version 5.0" on page 67

Restoring components and resources to Version 6.0

Restore components and resources that you have migrated from Version 6.0 products to their original state.

If you have migrated from WebSphere Message Broker Version 6.0 to Version 6.1, you can restore your installation to Version 6.0, subject to the following restrictions and procedures.

When you use the mqsimigratecomponents command, you must be logged in to a Version 6.1 environment.

See the mqsimigratecomponents command description for usage information and details of parameters and the format to use when specifying version numbers.

Restrictions

Message flows that you have deployed to a broker after migration to Version 6.1 might no longer work when you restore that broker to Version 6.0; always check logs to search for errors or warnings about message flows that the broker cannot start.

Source files in Message Broker Toolkit Version 6.1 are maintained in a different format to that used in Version 6.0. The files are migrated to the new format when you save them using the Message Broker Toolkit Version 6.1. After you have saved them, you can no longer use the resources in Message Broker Toolkit Version 6.0. For detailed information, see "Conditions for using migrated resources with previous versions of the Message Broker Toolkit" on page 6.

Migrating resources back to Version 6.0

The following sections describe how to restore the Message Broker Toolkit and your runtime components to Version 6.0.

Restoring the Message Broker Toolkit to Version 6.0

To restore the Message Broker Toolkit to Version 6.0:

1. Stop the Message Broker Toolkit Version 6.1.
2. Restore the Version 6.0 workspace from the backup that you took before migration.

   Ensure that the workspace directories include the .metadata directory, which contains information that is specific to the release of Eclipse on which the toolkit is based.
3. Restart Message Broker Toolkit Version 6.0.

Any changes that you made in Message Broker Toolkit Version 6.1 cannot be restored to Version 6.0.

Restoring runtime components to Version 6.0

Use the **-s** and **-t** parameters of the mqsimigratecomponents command to migrate components from Version 6.1 to Version 6.0.

- Specify the installed level of Version 6.1 for the source version parameter (**-s**), for example 6.1.0.0 for the GA level, or allow this to take the default value.
- Specify the appropriate level of Version 6.0 for the target version parameter (**-t**), for example 6.0.0.3 for Version 6.0, Fix Pack 3.

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

Restoring components on z/OS

To restore z/OS runtime components to Version 6.0:

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 6.0 copy that you backed up.

Restoring a broker on distributed systems

To restore a migrated broker to its Version 6.0 state:

1. Stop the Version 6.1 broker.
2. Restore the broker to Version 6.0 using the mqsimigratecomponents command, as shown in the following example:

   ```
   mqsimigratecomponents Broker -t 6.0.0.3
   ```

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 6.1.

   - On Linux and UNIX systems, reset the ODBCINI environment variable to point to the 32-bit version of the Version 6.0 odbc.ini file.

   - On Linux and UNIX systems, if you are using 64-bit execution groups, reset the ODBCINI64 environment variable to point to the 64-bit version of the Version 6.0 odbc.ini file (the file odbc64.ini).

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

4. Restart the broker using a Version 6.0 command window.

If you migrate to Version 6.1, deploy a message set to the Version 6.1 broker, and then migrate back to Version 6.0, Version 6.0 is unable to recognize the message set that was deployed by Version 6.1. In this case, any message sets that Version 6.0 is unable to use are deleted and a warning message is displayed for each message set, prompting you to redeploy it to Version 6.0 following successful migration.

Restoring a Configuration Manager on distributed systems
If you changed its queue manager when you migrated the Configuration Manager, you must keep the new queue manager if you restore using the mqsimigratecomponents command.

To restore a migrated Configuration Manager to its Version 6.0 state:

1. Stop the Version 6.1 Configuration Manager.

2. Restore the Configuration Manager to Version 6.0 using the mqsimigratecomponents command, as shown in the following example:

   `mqsimigratecomponents ConfigMgr -t 6.0.0.3`

3. Restart the Configuration Manager using a Version 6.0 command window.

Restoring a User Name Server on distributed systems
The User Name Server is unchanged between Version 6.0 and Version 6.1, therefore you do not have to restore this component.

If you choose to restore a User Name Server to Version 6.0, issue the mqsimigratecomponents command, as shown in the following example:

`mqsimigratecomponents UserNameServer -t 6.0.0.3`

Restoring components and resources to Version 5.0

If you have migrated from Version 5.0 to Version 6.1, you can restore your installation to Version 5.0, subject to certain restrictions.

Restrictions

- When you restore a broker from Version 6.1 to Version 5.0, message sets are deleted, therefore you must redeploy them. Running the mqsimigratecomponents command generates BIP warning messages that remind you to do this.
- Source files in Message Broker Toolkit Version 6.1 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 Broker Toolkit Version 6.1. After you have saved them, you can no longer use the resources in Message Broker Toolkit Version 5.0 or Version 5.1. For detailed information, see "Conditions for using migrated resources with previous versions of the Message Broker Toolkit" on page 6.
- When you have restored a broker, you must redeploy the bar files that you had previously deployed at Version 6.1. 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 Broker Toolkit and your runtime components to Version 5.0.

Restoring the Message Broker Toolkit to Version 5.0

To restore the Message Broker Toolkit to Version 5.0:
1. Stop the Message Broker Toolkit Version 6.1.
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 Broker Toolkit to this version.
3. Restart Message Broker Toolkit Version 5.0 with your Version 5.0 workspace.

Any changes that you made to resources in the Message Broker Toolkit Version 6.1 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 6.1 to Version 5.0.

- Specify the installed level of Version 6.1 for the source version parameter (**-s**), for example 6.1.0.0 for the GA level, 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 6.1 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
   ```

   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 6.1.

   - **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.

If you migrate to Version 6.1, deploy a message set to the Version 6.1 broker, and then migrate back to Version 5.0, Version 5.0 cannot recognize the message set that was deployed by Version 6.1. In this case, any message sets that Version 5.0 cannot use are deleted, and a warning message is displayed for each message set, prompting you to redeploy it to Version 5.0 following successful migration.

Restoring a User Name Server to Version 5.0

The User Name Server is unchanged between Version 5.0 and Version 6.1.

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 6.1, the DB2 database is neither changed nor deleted. Version 6.1 does not use DB2 for the configuration repository.

If you migrate from Version 6.1 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 6.1, 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 6.1.

If you deployed any broker configuration changes after migration to Version 6.1, these changes are lost when you restore the Configuration Manager to Version 5.0. You must rebuild your brokers by deleting them and redeploying

Migrating and upgrading **69**

them to maintain a consistent state. Check carefully that your Configuration Manager works correctly before deploying for the first time after migration to Version 6.1.

If you changed the Configuration Manager queue manager during migration to Version 6.1, 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
```

Part 2. Reference

Migration and upgrade

Consider the factors involved in the migration of components and resources from Version 5.0 or Version 6.0 to Version 6.1.

This section contains the following topics:

- "Supported migration paths"
- "Message flow migration notes" on page 74
- "Message set migration notes" on page 93
- "Equivalent z/OS customization variables in mqsicompcif and BIPEDIT" on page 99

Supported migration paths

You can migrate to WebSphere Message Broker Version 6.1 from previous versions of the product.

From ...	You can migrate to ...
WebSphere Message Broker Version 6.0[1]	• WebSphere Message Broker Version 6.1 • WebSphere Message Broker with Rules and Formatter Extension Version 6.1
WebSphere Message Broker with Rules and Formatter Extension Version 6.0[1]	• WebSphere Message Broker Version 6.1[2] • WebSphere Message Broker with Rules and Formatter Extension Version 6.1
WebSphere Event Broker Version 6.0[1 + 3]	• WebSphere Message Broker Version 6.1 • WebSphere Message Broker with Rules and Formatter Extension Version 6.1 • WebSphere Message Broker Version 6.0
WebSphere Business Integration Message Broker Version 5.0[4]	• WebSphere Message Broker Version 6.1 • WebSphere Message Broker with Rules and Formatter Extension Version 6.1 • WebSphere Message Broker Version 6.0 • WebSphere Message Broker with Rules and Formatter Extension Version 6.0
WebSphere Business Integration Message Broker with Rules and Formatter Extension Version 5.0[4]	• WebSphere Message Broker Version 6.1[2] • WebSphere Message Broker with Rules and Formatter Extension Version 6.1 • WebSphere Message Broker Version 6.0[2] • WebSphere Message Broker with Rules and Formatter Extension Version 6.0

From ...	You can migrate to ...
WebSphere Business Integration Event Broker Version 5.0[4]	• WebSphere Message Broker Version 6.1 • WebSphere Message Broker with Rules and Formatter Extension Version 6.1 • WebSphere Event Broker Version 6.0 • WebSphere Message Broker Version 6.0 • WebSphere Message Broker with Rules and Formatter Extension Version 6.0

Notes:

1. You can migrate Version 6.0 products at GA version or later.
2. This migration path is possible only if you have not used function unique to Rules and Formatter Extension in WebSphere Message Broker with Rules and Formatter Extension Version 6.0 or 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.
3. Version 6.0 is the last version of WebSphere Event Broker to be shipped. You can therefore migrate only to WebSphere Message Broker Version 6.1 or WebSphere Message Broker with Rules and Formatter Extension Version 6.1, or upgrade to WebSphere Message Broker Version 6.0 or WebSphere Message Broker with Rules and Formatter Extension Version 6.0.
4. You can migrate Version 5.0 products at Fix Pack 4 (5.0.0.4) or later. Earlier fix pack levels are not supported for migration to Version 6.1.

Message flow migration notes

This section provides reference information to help you to migrate message flows to WebSphere Message Broker Version 6.1.

It contains the following topics:
- "Migrating message mappings from Version 5.0" on page 75
- "Migrating a user-defined node" on page 79
- "Migrating a message flow that contains WebSphere MQ Everyplace nodes" on page 86
- "Migrating style sheets and XML files from Version 5.0" on page 81
- "Migrating a flow containing data definitions" on page 82
- "Migrating a flow containing HTTPRequest nodes" on page 85
- "Migrating a flow containing XMLTransformation nodes" on page 85
- "Migrating a flow that modifies input message trees" on page 93

You do not have to migrate mappings, style sheets, or XML files from Version 6.0 to Version 6.1.

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

Migrating message mappings from Version 5.0

Use the mqsimigratemfmaps command to migrate message mappings to the Version 6.1 format.

The mqsimigratemfmaps command creates Version 6.1 mapping files (.msgmap) from your Version 5.0 mapping files (.mfmap).

When you migrate message mappings from Version 5.0, read the restrictions that apply.

The following table lists the mapping functions that are supported in Version 5.0 but not supported in Version 6.1, and shows the error messages that you might see. Mappings that contain these Version 5.0 functions cannot be migrated to Version 6.1; you must re-create and redeploy these mappings using another node, such as a JavaCompute node. Alternatively, migrate as much of the mapping as possible using the migration command, view the error report to see details of the functions that could not be migrated, and create a new node that can execute those functions that were not migrated.

Supported in Version 5.0	Migration utility error message
Expressions that involve multiple instances of a repeating source element, for example: `src_msg.e[1] + src_msg.e[2] -> tgt_msg.e`	`Error:102: Unexpected index '2' encountered for target mappable 'e'. The expected index is '1'. Migration currently provides no support for expressions involving more than one instance of the same repeating-element.`
ESQL field references that contain the asterisk wildcard character "*". For example: `src_msg.e.* or src_msg.e.*[]`	`Error:130: ESQL field-reference 'src_msg.e.*' cannot be migrated. Migration currently provides no support for field-references containing '*'.`
Dynamic ESQL field references. For example: `src_msg.e.{'a' || 'b'}`	`Error:131: ESQL field-reference 'src_msg.e.{'a' || 'b'}' cannot be migrated. Migration currently provides no support for dynamic field-references.`
ESQL expressions that contain a reference to the temporary index-variable "#I". For example: `src_msg_e || "#I" -> tgt_msg.e`	`Error:128: ESQL expressions containing the variable '#I' anywhere other than the index of a repeating-element cannot be handled by the migration.`
Expressions within an index of a repeating element. For example: `src_msg.e[src_msg.a] or src_msg.e["#I" +5] or src_msg.e[< 3]`	`Error:116: ESQL field-reference 'src_msg.e[< 3]' cannot be migrated. Migration currently provides no support for indexes other than the variable '#I' and plain integer indexes.`
Aggregation functions MIN, MAX, and COUNT, used with the ESQL SELECT expression. For example: `SELECT MAX("#T".FIRSTNAME) FROM Database.CUSTOMER AS "#T" WHERE "#T".CUSTOMERID = 7`	`Error:135: The ESQL expression 'SELECT MAX("#T".FIRSTNAME) FROM Database.CUSTOMER AS "#T" WHERE "#T".CUSTOMERID = 7' could not be migrated. The expression contains syntax which has no direct equivalent in the new map-script language.`

Supported in Version 5.0	Migration utility error message

ESQL IN operator. For example:
 	
`src_msg.e IN (1, 2, 3)`
	`Error:135: The ESQL expression 'SELECT`
	`MAX("#T".FIRSTNAME) FROM Database.CUSTOMER AS "#T"`
	`WHERE "#T".CUSTOMERID = 7' could not be migrated.`

Restrictions on migrating message mappings

In certain scenarios, restrictions apply to the migration of .mfmap files from Version 5.0.

This topic explains why migration is not automatic in these situations, and provides instructions for how to complete a successful migration. This topic also provides information about restrictions that apply when you migrate submaps.

The programming model for message maps is different between Version 5.0 (where the file format is .mfmap) and Version 6.1 (where the format is .msgmap). Version 5.0 message maps have a procedural programming model, which is essentially an alternative ESQL, where you describe all the steps that are required to perform a transformation. Version 6.1 uses a declarative programming model, where you describe the result of the transformation, and the tools determine how to achieve that result.

Most migration failures result from message maps that contain too much information about the steps that perform the transformation, and not enough information about the desired result. For these message maps, migration is enabled by changing the .mfmap file so that specific "how to" sections are separated into an ESQL function or procedure that can be called by the message map. The .mfmap file calls the ESQL function instead of containing it as an expression. The mqsimigratemfmaps command then migrates the .mfmap file, but calls the ESQL function instead of logging a migration error.

A limitation is that ESQL (the run time for .mfmap and .msgmap files) cannot define functions that return complex element (or REFERENCE) values. The following procedure explains how to work around this complex element target limitation; in many cases, you must rewrite the map as an ESQL function. For more examples and information about calling ESQL from maps, refer to the following sample:

- Message Map sample

You can view samples only when you use the information center that is integrated with the Message Broker Toolkit.

1. Determine whether you can define an ESQL function for the .mfmap file.
   a. When the target value is a complex element, or in ESQL terms a REFERENCE, the individual mapping must be rewritten in the .msgmap file. Delete the mapping from the .mfmap file, and proceed to Step 4.
   b. Use a function for all other cases: CHAR string, numbers, date, and time. Proceed to Step 2.
2. Determine the source parameters and returns type for your function.
   a. For each source path in the mapping, there must be one parameter in the function or procedure. For a function, all parameters are unchangeable. The type of the parameter must match the source data type.
   b. The function return type is the ESQL data type identified above.

3. Update the .mfmap file to enable migration. Change the .mfmap file to invoke the function in the mapping, passing the source parameters to the function in the order in which they were listed in step 2a.
4. Re-run the mqsimigratemfmaps command to migrate the modified .mfmap file.
5. Repeat Steps 1 to 4 until no errors are reported in the migration log.
6. Start the Version 6.1 Message Broker Toolkit and open the migrated .msgmap file.
   a. For ESQL that is migrated as functions, there should be no errors.
   b. For complex element targets, rewrite the mapping using the Version 6.1 features.

The following examples illustrate migration of .mfmap files to .msgmap files.

- To migrate a multiple reference to a repeating source expression:

```
src_msg.e[1] + src_msg.e[2]
```

  compute the result in an ESQL function like:

```
CREATE FUNCTION addOneAndTwo(IN src_msg)
BEGIN
 RETURN src_msg.e[1] + src_msg.e[2];
END;
```

  In the .msgmap file, call the ESQL function addOneAndTwo using the parent element src_msg as a parameter.

- An expression that does not use element names:

```
src_msg.*
```

  or

```
src_msg.*[]
```

  can be processed using a function that takes the parent of the repeating field:

```
CREATE FUNCTION processAny(IN src_msg)
BEGIN
 DECLARE nodeRef REFERENCE TO src_msg.e.*;
 DECLARE result <dataType> <initialValue>;
 WHILE LASTMOVE nodeRef DO
  --expression goes here
  SET result = result;
 END WHILE;
 RETURN RESULT;
END;
```

  In the .msgmap file, call the ESQL function processAny using the parent element src_msg as a parameter.

- Expressions that dynamically compute element names:

```
src_msg.{'a' || 'b'}
```

  can be processed by ESQL functions that process the parent of the repeating field:

```
CREATE FUNCTION processDynamicName(IN src_msg)
BEGIN
 RETURN src_msg.{'a' || 'b'};
END;
```

  In the .msgmap file, call the ESQL function processDynamicName using the parent element src_msg as a parameter.

- Expressions that use the select MIN, MAX, and COUNT functions:
```
SELECT MAX("#T".FIRSTNAME)
 FROM Database.CUSTOMER AS "#T"
 WHERE "#T".CUSTOMERID = custId
```

  can be processed by ESQL functions that process the parent of the repeating field:
```
CREATE FUNCTION processMAX(IN custId)
BEGIN
 RETURN
 SELECT MAX("#T".FIRSTNAME)
  FROM Database.CUSTOMER AS "#T"
  WHERE "#T".CUSTOMERID = custId
END;
```

  In the .msgmap file, call the ESQL function processMAX using the element custId as a parameter.
- .mfmap files that use mfmap index variables in expressions:
```
e || "#I"
```

  must be rewritten entirely in ESQL. By definition, there must be a complex repeating parent element, and this is not supported by ESQL functions.
- Expressions that use source expressions to compute values:
```
src_msg.e[src_msg.a]
```

  must be rewritten using if rows, msgmap:occurrence() functions, and ESQL functions:
```
for src_msg.e
 if
  condition msgmap:occurrence(src_msg/e) = src_msg/a
```
- For expressions that use index expressions to compute values:
```
src_msg.e["#I" +5]
src_msg.e[< 3]
```

  the entire .msgmap file must be rewritten in ESQL, because the .msgmap files do not support indexed access to repeating fields.
- .mfmap files that use ROW expressions to compute values:
```
src_msg.e IN (1, 2, 3)
```

  must be rewritten in ESQL, because .msgmap files do not support ESQL ROW expressions.

Restrictions on migrating maps that call ESQL

If there is a mismatch between the case that has been in the ESQL call in the message map, and the name of the routine defined in the ESQL file, an error is produced during migration of the message map. To prevent an error occurring during migration, ensure that the ESQL call in the message map uses the same case as the ESQL defined in the routines in the ESQL file. Alternatively you can manually edit the message map after migration to call the ESQL routine with matching case.

Restrictions on migrating submaps

In Version 5.0 message maps, any complex element type can be a root of a submap. However, in Version 6.1, only a global element or a global attribute can be

the root of a submap. When a Version 5.0 message map with a call to a submap with a non-global element as the map root is migrated, the submap is not migrated as a standalone submap. Instead, the call to the submap in the main message map is replaced by the migrated content of the submap. Alternatively, if the submap has a global element as the map root, the submap is migrated to a standalone Version 6.1 submap instead.

For Version 6.1, you must define reusable schema structures as global elements and types. If you have Version 5.0 submaps that use local elements, you must change the schema to add definitions of global elements for the local elements, and then use the new schema after migration. If the new global elements have the same name and type as the local elements, the Version 5.0 submaps do not need to be changed.

You must qualify a local element in a Version 5.0 submap with a namespace to ensure its successful migration to Version 6.1, because the global element that replaces it after migration must be qualified by the namespace. If your submap contains local elements, you must re-create the submap and re-create the call to the submap from the main message map.

The following table shows differences between the features that are supported in a submap for Version 5.0 and Version 6.1.

Version	Supported feature
Version 5.0	global elements and global attributes as map source
	global elements and global attributes as map target
	local elements and local attributes as map source
	local elements and local attributes as map target
Version 6.1	global elements, global attributes, and global types as map source
	global elements and global attributes as map target

Migrating a user-defined node

Migrate your user-defined nodes to Version 6.1.

The tasks depend on the version that you are migrating from:
- "Migrating a user-defined node from Version 6.0 and Version 5.1"
- "Migrating a user-defined node from Version 5.0" on page 80

Migrating a user-defined node from Version 6.0 and Version 5.1

To migrate from Version 6.0 or Version 5.1 to Version 6.1:
1. Import the user-defined node project into the Version 6.1 Message Broker Toolkit.
2. Select your user-defined node project in the Package Explorer, and click **Project** → **Clean Project**.

3. If you have created new Version 6.1 brokers, copy the runtime code to each new broker system.
   - For C user-defined extensions, compile the code for the target operating system. Save the LIL or JAR file in the directory that you specified for the -l parameter on the mqsicreatebroker or mqsichangebroker command.

     Also store the PDB file that corresponds to the LIL file in the same directory. The PDB file provides symbolic information that is used in stack diagnostic information in the event of access violations or other software exceptions.
   - For Java user-defined nodes, save the LIL or JAR file in the directory that you specified for the -l parameter on the mqsicreatebroker or mqsichangebroker command.

Migrating a user-defined node from Version 5.0

To migrate a user-defined node from Version 5.0 to Version 6.1, you must migrate both toolkit and runtime components.

Complete the following steps to migrate a user-defined node:
1. "Migrating the Message Broker Toolkit representation of the user-defined node"
2. "Migrating the user-defined node runtime code"

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

To migrate from the Version 5.1 Message Brokers Toolkit to the Version 6.1 Message Broker Toolkit:
1. Import the user-defined node project into the Version 6.1 Message Broker Toolkit.
2. Select your user-defined node project in the Package Explorer, and click **Project → Clean Project**.

To migrate from the Version 5.0 Message Brokers Toolkit to the Version 6.1 Message Broker Toolkit:
1. Import the user-defined node project into the Version 6.1 Message Broker Toolkit.
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 6.1 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 Broker Toolkit, and you have not yet deployed to it, you can add the LIL, PDB, and JAR files to your chosen directory.
   - If you have already deployed to the execution group that you want to use, add the LIL, PDB, and JAR files 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 style sheets and XML files from Version 5.0

You might have to make changes to message flows and to the style sheet and XML files that they use when you migrate from Version 5.0 to Version 6.1.

If you do not plan to include the style sheets and XML files with the message flows that use them in the broker archive file that you deploy, and your XSLTransform nodes reference style sheets with partially-qualified paths (relative referencing), you might need to migrate both the message flows and the style

sheets. If your XSLTransform nodes reference style sheets with fully-qualified paths (absolute referencing), migration is not necessary.

The XSLTransform node has been renamed in Version 6.1. In Version 5.0 it is called XMLTransformation. The node behavior, properties, and terminals are unchanged. You do not have to change message flows that include this node, and you can deploy them unchanged to Version 6.1 brokers.

You can continue to deploy your style sheets and XML files manually into the broker file systems, but you must store relatively-referenced style sheets and XML files (and their dependants) in the *broker work path*/{XSL,XML}/external directories.

In Version 5.0, any relative references to manually-deployed style sheets are relative to the location of the file directory where the broker is started. In later versions, any relative references to manually-deployed style sheets are relative to the directory *broker work path*/{XSL,XML}/external on the broker file system. You must therefore complete the following actions:

1. Move the relatively-referenced style sheets (and their descendants), to the correct directory structure under *broker work path*/{XSL}/external.
2. Move all their dependent XML files to the correct directory structure under *broker work path*/{XML}/external.

For example, a relatively-referenced principal style sheet, a/b.xsl, must now be stored as *broker work path*/{XSL}/external/a/b.xsl on the broker file system. An XML file that is referenced by a/b.xsl as c/d.xml must be stored as *broker work path*/{XML}/external/a/c/d.xml.

An embedded style sheet is handled in the same way as a file that is loaded from the root of the broker's deployed storage, therefore move any descendant style sheets or XML files of embedded style sheets to the *broker work path*/{XSL,XML}/external tree. If you cannot move the files, you can reference them with a fully-qualified path (absolute reference).

Migrating a flow containing data definitions

When you migrate a message flow that uses data definitions, Version 6.1 converts and replaces those definitions.

WebSphere Message Broker Version 6.1 manages data definitions in a data design project. A single file in the project (a .dbm file) represents the database. This file replaces the set of .xmi files that you created for each data definition in the Version 5.0, Version 5.1, or Version 6.0 workbench.

How you start migration depends on the version that you are migrating from:
- If you are migrating from Version 6.0, migration is performed only when you rebuild the message flow project in the workbench. Therefore you must start a Version 6.1 workbench session to migrate the data definitions. See "Migrating data definitions in the workbench" on page 83.
- If you are migrating from Version 5.0 or Version 5.1, use the mqsimigratemfmaps command.

  Use this command if you have mappings, database definitions, or both, to migrate. See "Migrating data definitions using the mqsimigratemfmaps command" on page 84.

If you prefer, you can open a Version 6.1 workbench session and rebuild the message flow projects that are affected to migrate the data definitions. However you must use the mqsimigratemfmaps command to migrate mappings.

Because the project files are changed by the migration process, and the old format files are deleted, back up your resources before you start to use existing resources in the Version 6.1 workbench.

When migration is complete, you can view data design projects in the Broker Application Development perspective, and you can view the contents of the .dbm files in the Data Project and Database Explorer views.

If a data definition is not migrated successfully, the database might no longer be supported. Check the version of the database that is referenced by the data definition against the Supported databases, and upgrade your database if necessary.

Migrating data definitions in the workbench

Complete the following steps:

1. Start the Version 6.1 workbench and switch to the Broker Application Development perspective.
2. Import the message flow that contains data definitions in .xmi files. You are asked to confirm that you want migration to be performed.
3. Select **Do not show this message again** if you want all projects to be migrated automatically when you import or rebuild them, and click **Yes** to continue with migration. The project is rebuilt, all data definitions are converted to the .dbm file format, and all database references from ESQL modules and mappings are replaced.

   To reset the option that suppresses the confirmation dialog, click **Windows** → **Preferences**, open **Broker Development**, and click **Database Definition**. Change the setting on this panel.
4. Clean all your projects after migration:
   a. Click **Project** → **Clean**.
   b. Select **Clean all projects**.
   c. Click **OK**.
5. Check the results of the migration. Errors and warnings are displayed in the Tasks view.

If you decide not to continue with migration, some errors might be generated in ESQL modules (.esql files) and in mappings (.msgmap files) that refer to databases. You must migrate the database resources in the project before you can deploy the project. You can start migration later by rebuilding the message flow project, or right-clicking the Database Connections folder, and selecting **Database Definition Migration**. (The Database Connections folder appears in the toolkit only if you have not migrated all your database definitions, and it is removed when all migration has completed.)

See "How the migration works" on page 84 for details about changes that are made to the contents of the workspace.

Migrating data definitions using the mqsimigratemfmaps command

If you are migrating from Version 5.0, use the mqsimigratemfmaps command to migrate message flow projects that contain mappings, or data definitions, or both.

Complete the following steps:

1. Open a command window that is configured for your Version 6.1 installation.

   - **Linux** On Linux on x86, run the mqsiprofile command.

   - **Windows** On Windows, click **Start → Programs → IBM WebSphere Message Brokers 6.1 → Command Console**.

2. Run the mqsimigratemfmaps command. Specify the parameters to perform the operation that you want. For example, you can migrate all projects in the source directory, or you can migrate one or more specific projects. You can also specify the path and name of the log file in which progress is reported.

3. Check the contents of the log file. If you have not specified an alternative log file, the migration results are recorded in file mqsimigratemfmaps.report.txt in the folder from which you ran the command.

4. Start the Version 6.1 workbench and switch to the Broker Application Development perspective to view the migrated resources.

5. Clean all your projects after migration:

   a. Click **Project → Clean**.

   b. Select **Clean all projects**.

   c. Click **OK**.

See "How the migration works" for details about that changes that are made to the contents of the workspace.

How the migration works:

The migration is handled in the following way:

- A new data design project is created if database definitions are found in the project. The first data design project name is set to the message flow project name, appended with _DDP, for example FlowProject1_DDP.

  Each database definition is converted to Version 6.1 format (that is, the set of .xmi files is converted and stored as a single file called *database_name*.dbm). Multiple unique database definitions are converted and stored in a single data design project.

  You can rename the .dbm file, or the data design project.

- If you have created duplicate database definitions for the same database in the message flow project, another data design project is created for each definition.

  In Version 5.0, Version 5.1, and Version 6.0, you can create multiple database definitions for the same database within a single message flow project. In Version 6.1 you can have only one definition, for any one database, to ensure that mappings are unambiguous.

  You can create separate data definitions for the same database in separate data design projects, but you can refer to only one of these definitions from a single message flow project.

  The .xmi files are converted to a single .dbm file and each duplicate definition is stored in a separate data design project. The names of additional data design

projects are also based on the name of the message flow project, and are appended with _DDP2, _DDP3, and so on.

You might have to modify your message flow projects to ensure that all database references are to the same definition of the database.

- All old database definition files are deleted from the message flow project and from the file system.
- The first data design project that is created is added as a project reference to the message flow project.

If data design projects have been created for duplicate database definitions, you cannot refer to these projects from the same message flow project.

If you change the name of the data design project, you must modify the project reference in the message flow project.

- During conversion, information and errors are recorded in a log. Errors are also displayed on a dialog to indicate that migration has failed for a definition, and the cause of the failure. After migration, view the log file migratedbdefinitions.log in the .metadata/.plugins/com.ibm.etools.mft.rdb directory.

The log contains details of the data design projects that have been created, the .dbm files that have been added to those projects, and the .xmi files that have been deleted.

If you migrate these resources at a later date, for example when you import a different message flow project, the log records are appended to the existing content.

Migrating a flow containing HTTPRequest nodes

When you migrate a message flow containing HTTPRequest nodes, check how Version 6.1 performs message validation in these nodes.

In WebSphere Message Broker Version 6.1, message validation for HTTPRequest nodes has changed. The validation options that you can specify on an HTTPRequest node apply only to the response message that is received by the node from the remote server when an HTTP request is made. In earlier versions, these options were also applied to the message received by the node's In terminal.

The message received at the In terminal is no longer validated, however, validation of the outbound (request) message can still be performed. Set validation options on a preceding Compute node, or include a Validate node. The HTTPRequest node uses the validation options associated with the message that is received at the In terminal to validate that message when it is sent out.

Migrating a flow containing XMLTransformation nodes

When you migrate a message flow containing XMLTransformation nodes, check how Version 6.1 uses the node properties and search priorities.

Prior to Version 6.1, the XSLTransform node was known as the XMLTransformation node. When you migrate a flow containing XMLTransformation nodes to Version 6.1, consider the following enhancements:

- You can no longer set the style sheet search and character set information search priorities in the node properties in the workbench. The following fixed search priorities are adopted. However, search priorities set on existing flows are still respected.

– The node searches for the name of the style sheet to be used by interrogating each of the following items in turn:
1. The input message. The node searches the message XML data for stylesheet location information. For example, if the XML data contains this entry:

```
<?xml-stylesheet type="text/xsl" href="foo.xsl"?>
```

the node uses "foo.xsl" as the stylesheet name.
2. The LocalEnvironment. If no stylesheet name is found in the input message, the node searches the LocalEnvironment associated with the current message for stylesheet information stored in an element called XSL.StyleSheetName.

In versions prior to Version 6.1, element ComIbmXslXmltStylesheetname is used for the name of the style sheet, so the current node checks both elements. If both are present, the value in XSL.StyleSheetName takes precedence.
3. The node's properties. If no stylesheet name is found in the input message or LocalEnvironment, the node uses the Stylesheet Name and Stylesheet Directory properties to determine the correct values.

– The node searches for the character set to use for the output message by interrogating each of he following items in turn:
1. The LocalEnvironment. The node searches the LocalEnvironment associated with the current message for character set information stored in an element called XSL.OutputCharSet; for example, to encode the output of the transformation as UTF-16, enter the value 1200 as a string in this element.

In versions prior to Version 6.1, element ComIbmXslXmltOutputcharset is used for the output character set, so the current node checks both elements. If both are present, the value in XSL.OutputCharSet takes precedence.
2. The node's properties. If no character set information is found in the LocalEnvironment, the node uses the Output Character Set property to determine the correct value.

If the node cannot determine the output character set from either of these two sources, either because no value is set or the selection priorities are set to zero, the default value of 1208 (UTF-8) is used.

• You can set additional node properties, or use additional LocalEnvironment overrides, to specify the message domain, message set, message type, and message format of the output message. If you have included a ResetContentDescriptor node to set these values before passing the message to the XSLTransform node, you can now optimize the message flow by removing the ResetContentDescriptor node.

Migrating a message flow that contains WebSphere MQ Everyplace nodes

Support for WebSphere MQ Everyplace in WebSphere Message Broker Version 6.1 differs from that provided in WebSphere Business Integration Message 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)

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 6.1 configuration

Using message flows that contain MQeInput and MQeOutput nodes in Version 6.1 is deprecated. The behavior that is described here is intended only for when you are deploying from Version 6.1 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 6.1, 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 6.1 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 6.1 broker, change the Destination Mode property to Reply to Queue, and then redeploy the modified flows. You can use the Message Broker Toolkit at either Version 6.1, or at Version 5.0 (or Version 5.1) to perform the redeployment.

If you want to use WebSphere MQ Everyplace connections with Version 6.1, see the subsequent topics for more details, starting with "Designing MQe connections" on page 88.

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 6.1.

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 86
2. Designing MQe connections (this topic)
3. "Deploying an MQe message flow" on page 89
4. "Configuring after MQe message flow deployment" on page 91

This topic describes the connection options between WebSphere MQ Everyplace device clients and WebSphere Message 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:
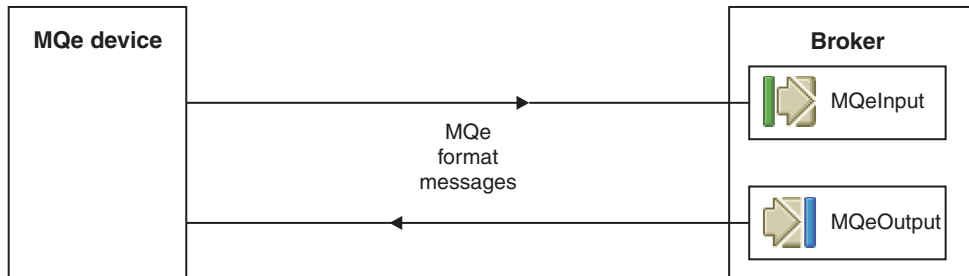
   - Through an MQe gateway:
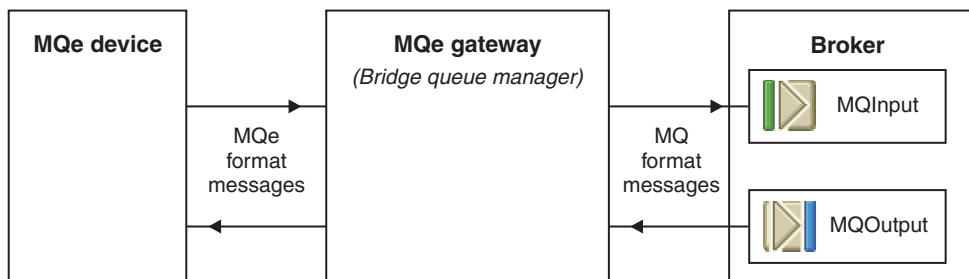
2. The connection option for Version 6.1:
   - Through an MQe gateway:
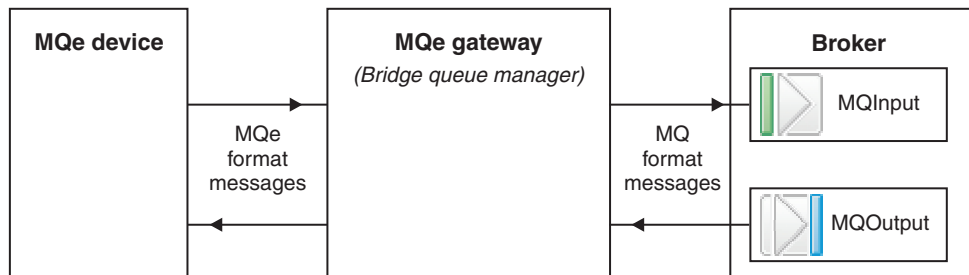
The following table lists connection options between WebSphere MQ Everyplace device clients and WebSphere Message Broker.

Connection considerations	Version 5.0	Version 6.1
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 6.1.

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 86
2. "Designing MQe connections" on page 88
3. Deploying an MQe message flow (this topic)
4. "Configuring after MQe message flow deployment" on page 91

Using message flows that contain MQeInput and MQeOutput nodes in Version 6.1 is deprecated. The behavior that is described here is intended only for when you are deploying from Version 6.1 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 6.1	Not supported. Import your existing message flow into the Message Broker Toolkit Version 6.1 and deploy from there.[1]
Version 6.1	Version 5.0	Works as before.[2]

Deploy from	Deploy to	Result
Version 6.1	Version 6.1	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 6.1, 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 6.1 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" on page 91).

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 Message 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 6.1.

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 86
2. "Designing MQe connections" on page 88
3. "Deploying an MQe message flow" on page 89
4. Configuring after MQe message flow deployment (this topic)

Using message flows that contain MQeInput and MQeOutput nodes in Version 6.1 is deprecated. The behavior that is described here is intended only for when you are deploying from Version 6.1 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 6.1 the nodes are converted to MQInput and MQOutput nodes (as described in "Deploying an MQe message flow" on page 89).

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:
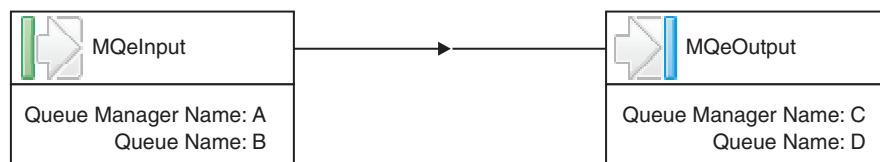
The following diagram shows this message flow connected directly to MQe devices, as it would be used on a Version 5.0 broker:

MQe device

- Queue manager: A
  - Queue: B

MQe format →

Broker

MQeInput

Queue Manager Name: A
Queue Name: B

MQe device

- Queue manager: C
  - Queue: D

← MQe format

MQeOutput

Queue Manager Name: C
Queue Name: D
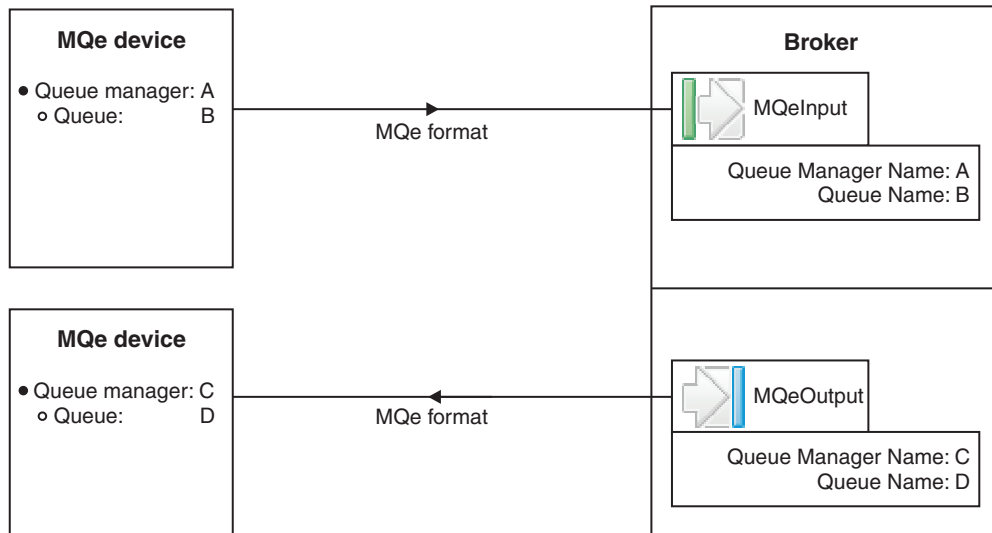
When you deploy this message flow to a Version 6.1 broker, create a configuration as shown below:

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

Listener

← MQ 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

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.

Migrating a flow that modifies input message trees

You might see a change in behavior when you migrate message flows that modify input message trees.

When you migrate message flows from previous versions of WebSphere Message Broker that have been incorrectly modifying input message trees in ESQL nodes or user-defined nodes, you might see a change in behavior because the changes are now reflected in a bit stream that is written from that tree.

Message set migration notes

Read these notes to help you to migrate message sets to WebSphere Message Broker Version 6.1.

This topic contains the following sections:
- "Migrating from Version 6.0"
- "Migrating from Version 5.0" on page 94

If you are using the Message Broker Toolkit Version 5.1, replace all references in this topic to "Version 5.0" with "Version 5.1".

Migrating from Version 6.0

To migrate message sets from Version 6.0 to Version 6.1, no migration commands are necessary. Message Broker Toolkit Version 6.1 can read the content of a Version 6.0 message set project and automatically converts it to Version 6.1 format when you modify and save it for the first time.

For guidance about changes to the behavior of parsers, and for general migration information, see the following sections:
- "Behavioral changes in the XMLNSC parser" on page 94
- "Behavioral changes in the MRM parser" on page 95
- "General migration information" on page 97

Migrating from Version 5.0

To migrate message sets from Version 5.0 to Version 6.1, no migration commands are necessary. Message Broker Toolkit Version 6.1 can read the content of a Version 5.0 message set project and automatically converts it to Version 6.1 format when you modify and save it for the first time.

For guidance about changes to the behavior of parsers, and for general migration information, see the following sections:
- "Behavioral changes in the MRM parser" on page 95
- "General migration information" on page 97

Behavioral changes in the XMLNSC parser

The following items are relevant when you migrate from Version 6.0 only:
- In Version 6.0, the attributes of an XML element appear in the message tree that is created by the XMLNSC parser in the order in which they appear on the element in the XML document. In Version 6.1, the XMLNSC parser always places namespace attributes ahead of other attributes when it creates the message tree. XML attributes are inherently unordered, therefore do not write message flow logic that assumes attribute position.
- In Version 6.0, the XMLNSC parser accepts the EBCDIC NEL as an XML line feed character on z/OS only. In Version 6.1, NEL is accepted on all platforms.
- In Version 6.0, the XMLNSC parser ignores a node's Validate property setting. In Version 6.1, the XMLNSC parser honours the Validate property setting. If you have set this property to Content or Content And Value, the parser validates the message against the message set defined by the Message Set property. Therefore, if you have accidentally set the Validate property in Version 6.0 to Content or Content And Value, the message flow throws an exception after migration to Version 6.1. To correct this situation, set the Validate property to None.
- In Version 6.0. the XMLNSC parser uses the default ESQL format "YYYY-MM-DD hh:mm:ss" when it writes a field of type TIMESTAMP or GMTTIMESTAMP. This format does not comply with the XML Schema 1.0 specification. In Version 6.1, the XMLNSC parser uses the format "YYYY-MM-DD'T'hh:mm:ss.sss", which complies with the XML Schema 1.0 specification. If you require another format to be written, you must cast the TIMESTAMP or GMTTIMESTAMP to a CHARACTER. (If the element was originally parsed by the XMLNSC parser and had a time zone indicator, the time zone indicator is also written to the output message field.)
- In Version 6.0, the XMLNSC parser uses the default ESQL strings "TRUE" and "FALSE" when it writes a field of type BOOLEAN. This use does not comply with the XML Schema 1.0 specification. In Version 6.1, the XMLNSC parser writes "true" and "false", which complies with the XML Schema 1.0 specification. If you require strings other than "true and "false" to be written, cast the BOOLEAN field to a CHARACTER.
- In Version 6.0, the XMLNSC parser does not validate XML documents against XML Schema generated from a message set. In Version 6.1, the XMLNSC parser has this capability. When you add a message set to a bar file, if the Message Domain property of the message set is set to XMLNSC, an XML Schema is generated for each message definition file in the message set, for potential use by the XMLNSC parser. These XML Schema are checked by the XMLNSC parser when the bar file is deployed. However, the checks are stricter than the checks that are carried out by the message definition file editor, and the deployment

might fail. If the deployment fails, you must correct the message definition file to comply with XML Schema rules. If the message set contains one of the IBM-supplied SOAP Envelope message definition files, re-import using the New Message Definition File From IBM-supplied Message wizard.

Behavioral changes in the MRM parser

The following changes are implemented in Version 6.1 and are relevant when you migrate from Version 5.0 and Version 6.0:

- In earlier releases, the MRM parser TDS physical format Data Element Separations Tagged Fixed Length, Tagged Delimited, and Tagged Encoded Length write out any xsi:type attributes they encounter in the message tree, using a tag name of "type".

  In Version 6.1, the TDS physical format assumes that all xsi:type attributes are XML-specific, and does not write them out.

- In earlier releases, the MRM TDS physical format Data Element Separations Fixed Length, Fixed Length AL3, and Tagged Fixed Length treat an element with a TDS Length of zero as fixed length when parsing, but as variable length when writing. Therefore, any data in the message tree for such an element is written to the bit stream without an exception being thrown.

  In Version 6.1, the element is treated as fixed length when parsing and writing. Therefore, any data in the message tree for such an element causes a BIP5436 exception to be thrown, unless you have set the TDS property Truncate on output, in which case nothing is written.

- In earlier releases, exception BIP5374E is thrown when a missing mandatory repeating element (minOccurs > 1) is detected during validation of an MRM message.

  In Version 6.1, two different conditions are detected during validation:

  – A mandatory element that is repeating but has no instances in the message tree. In this case, exception BIP5378E is thrown, which provides details of the elements that exist before and after the expected location of the missing element. Because of this change, you might want make changes or enhancements to any automated error processes that you have in place.

  – A mandatory element that is repeating and has the incorrect number of instances in the message tree. In this case, exception BIP5374E is thrown as before, giving details of the existing and the expected instances of the element.

The following changes are implemented in Version 6.1 and are relevant when you migrate from Version 5.0 only:

- XML physical format. Version 6.1 is sensitive to xsi:type attributes in the XML document and modifies its behavior accordingly. xsi:type attributes are no longer treated as self-defining attributes, therefore they appear in the message tree with the name 'type' instead of '@type'. If your message flow logic is sensitive to xsi:type attributes in the message tree, change your message flow to comply with the new behavior. To retain pre-Version 6.1 logic in your message flows, set the environment variable MQSI_IGNORE_XSI_TYPE to any value, and the pre-Version 6.1 behavior is adopted.

- XML physical format. DOCTYPE information in an XML document does not appear in the logical message tree when parsed, but it is preserved from input document to output document because the MRM keeps a copy of the DOCTYPE internally. Prior to Version 6.0, this data is an exact copy of the bit stream. In

Version 6.1, the copying process causes some formatting white space to be lost. The following example shows an element declaration in an input DOCTYPE:

```
<!ELEMENT e0 (e1|e2)+ >
```

In the output message, the element declaration has changed:

```
<!ELEMENT e0 (e1|e2)+>
```

The new behavior is consistent with the way that the XML physical format processes white space in all other XML constructs.

- TDS physical format. For a group with logical property Composition set to Choice and TDS property Data Element Separation set to Fixed Length, the parser always assumes that the length of the choice data is that of the longest child of the group, and always reads and writes that number of characters. Ensure that your messages comply with this assumption, otherwise the parser will treat data that follows the choice as part of the choice. Prior to Version 6.0, the parser does not enforce this rule.

- TDS physical format. If TDS element property Encoding Null is set to NullPadFill, it is only acted upon if the corresponding Length property is being used actively when parsing or writing. Prior to Version 6.0, NullPadFill is acted upon regardless of whether the Length property is being used actively.

- TDS physical format. TDS property Data Element Separation is set to All Elements Delimited or Variable Length Elements Delimited. A complex element is now always treated as having variable length, regardless of its own Data Element Separation setting. Therefore, a Delimiter is expected in the message bit stream between the complex element and any subsequent element, and a Repeating Element Delimiter is expected between all instances of a complex element, even if the complex element has a fixed length. Prior to Version 6.0, the parser does not enforce this rule, and for Variable Length Elements Delimited, the writer does not output delimiters when the length of the complex element was known.

- TDS physical format. TDS property Data Element Separation is set to Use Data Pattern. If the regular expression that is specified for a data pattern returns a zero length match, it is treated as an element with a zero-length value. Prior to Version 6.0, a zero-length match is treated as an element being omitted.

- TDS physical format. TDS property Data Element Separation is set to Tagged Delimited. An extraneous delimiter that occurs after the last child of the group in the bit stream is no longer tolerated and causes a parsing exception. If this is a problem, consider modeling the extraneous delimiter using the TDS Group Terminator property. Prior to Version 6.0, the parser does not enforce this rule.

- TDS physical format. TDS property Data Element Separation is set to Tagged Delimited. In Version 6.1, when parsing a Tagged Delimited group, the parser attempts to make sense of groups where the members appear out of order in the message bit stream, even if the group Composition property is set to Sequence or OrderedSet. However, if the group contains either an embedded message or a complex element or group that cannot be identified from the bit stream, all the members of the group must appear in the bit stream in the same order that they are defined in the model. If they appear out of order, the group is not parsed correctly and has unpredictable results. One symptom of this condition is the appearance of self-defining elements in the message tree, caused by a failure to match an element to the model.

  A specific example of this condition is where your message contains an embedded message and you are using either the Message Key or Message Identity technique to identify the embedded message. If the element that is

providing the message key or message identity value fails to be matched with the model, the parser does not know whether to interpret its value as a message key or message identity.

Prior to Version 6.0, the parser attempts to make sense of all out-of-order Tagged Delimited groups, with a consequent reduction in performance. In Version 6.1, if this behavior is a problem, consider modeling the unordered content of the group as an embedded child group with Composition set to UnorderedSet.

A complex element or group can be identified from the bit stream if it provides a group indicator, a tag, or a data pattern, or if its child members provide a group indicator, tag, or data pattern.

Despite its name, under some circumstances members of a Tagged Delimited group do not need to provide a tag; specifically, if the member is an embedded message or is a complex element or group.

- TDS physical format. The TDS Data Element Separation property is set to Tagged Encoded Length. When writing, the encoded length is written as the number of characters in the data, to match the interpretation when parsing. Prior to Version 6.0, the encoded length is written as the number of bytes in the data, which does not match the interpretation when parsing if the data is not SBCS.

- TDS physical format. When data is written, if the number of repeats of an element or group is fixed in the model but the actual occurrences in the logical message tree exceed the number of repeats, and the Data Element Separation property is not one of the "Tagged" separations, the extra occurrences are discarded and do not appear in the output message. Prior to Version 6.0, the extra occurrences appear in the output message. If the extra repeats are expected to appear in the output message, specify maxOccurs=-1 to indicate that the number of repeats is unbounded.

- All physical formats. When writing dateTime fields that have a Format property of I, the output form depends on the logical type of the field, as described in DateTime as string data. Prior to Version 6.0, the output form is incorrectly the full yyyy-MM-dd'T'HH:mm:ssZZZ form.

- Validation of input and output messages has been improved to provide additional detection of invalid messages. In Version 6.1, a message might therefore be flagged as invalid when it was incorrectly not flagged in previous versions.

General migration information

The following items are relevant when you migrate from Version 6.0 and Version 5.0:

- The XML parser is deprecated and might be withdrawn in a future release. Existing message flows that use the XML parser continue to operate in Version 6.1. Use the XMLNSC domain for most new message flows that need to parse or write XML documents. For more information, see Which XML parser should you use?.

- The IDOC domain and parser are deprecated, and might be withdrawn in a future release. Existing message flows that use the IDOC parser continue to operate in Version 6.1. Use the MRM domain TDS physical format for new message flows that need to parse or write SAP ALE IDocs in text format that originate from the WebSphere MQ link for R3.

- The location and labeling of some MRM TDS properties have been updated; no behavioral changes are associated with these changes.
  - The Repeating Element Delimiter property has moved into a box labeled "Occurrences", to keep all repeating properties together.

- The Virtual Decimal Point and Precision properties are moved below the "Sign" group of properties, to align them with CWF properties.
- The Length Reference property was not used for local groups, group references, and complex elements, and is removed from their property sheets.
- The local group property sheet now has two boxes labeled "Field Identification", to distinguish the Data Pattern property from the rest, because Data Pattern applies not to the group itself, but to its inclusion in its parent.
- The message set Boolean representation properties are now displayed with prefix "Text" to distinguish them from new binary Boolean representation properties.
- The Physical Type property enumeration value Character is renamed Text.
- The Physical Type property enumeration value Messaging Standard Alternate is renamed TLOG Specific.
- The Trim Fixed Len String property is renamed Trim On Input, and has moved into a new box labeled "Fixed length strings".
- The Derive default length from logical type property has moved into a new box labeled "General settings".
- The location and labeling of some MRM CWF properties have been updated; no behavioral changes are associated with these changes.
  - The Format property has moved to be last in the "Physical representation" section, to align it with TDS properties.
  - The element property sheet now has a section labeled "Numeric representation" section which contains the Sign group of properties and Virtual Decimal Point, to align these properties with TDS properties.
  - The Length Count property is renamed Length, to align this property with TDS properties.
  - The String Justification property is renamed Justification, to align this property with TDS properties.
  - The Sign EBCDIC Custom property is renamed Sign EBCDIC Custom Overpunched, to align this property with TDS properties.
  - The Sign Orientation property enumeration values Leading and Trailing are renamed Leading Overpunched and Trailing Overpunched, to align these properties with TDS properties.
  - The message set byte order properties are now displayed with the prefix "Default" to align these properties with TDS properties.

The following items are relevant when you migrate from Version 5.0 only:

- The CWF property Repeat Count has been superseded by the logical property Max Occurs, bringing the CWF physical format in line with TDS and XML physical formats, which already use Max Occurs to determine the number of repeats. A warning appears in the Problems view of the message editor for each element or group that had a CWF Repeat Count value set. Right-click the warning and click **Quick Fix** to resolve the issue.

  In Version 5.0, if the CWF property Repeat Count is not set *and* Min Occurs is not equal to Max Occurs, the number of repeats is taken to be one. In Version 6.0 and Version 6.1, the number of repeats is taken to be Max Occurs. A warning cannot be generated for this situation. Such a message model is not created by the COBOL or C importers, therefore it can occur only if you have created a CWF message model using the Version 5.0 message editor.

- The TDS physical format prior to Version 6.0 included embedded message identification by Message Key. The Message Key technique has been deprecated

and is superseded by a technique called Message Identity. A warning appears in the Problems view of the message editor for each message that has a TDS Message Key value, and for each element or attribute that has TDS Interpret Element Value property set to Message Key. Right-click the warning and click **Quick Fix** to resolve the issue.

Continue to use TDS Message Key if the message set will ever be deployed to a Version 5.0 broker, because these brokers do not support the Message Identity technique of embedded message identification.

- The TDS physical format property Data Pattern is now validated by the Message Broker Toolkit to ensure that it is a valid XML Schema regular expression. Errors appear in the Problems view of the message editor, and you must correct them manually using the editor. In particular, some errors are issued against the XML Schema specification concerning the escaping of meta-characters, which can cause validation errors to appear. To correct the errors, you might need to escape the hyphen character (-) or the brace characters ({) and (}), using the backslash character (\); for example \{ or \-. If you receive such errors when using an IBM-supplied message set, contact IBM® to obtain a corrected version of the message set.

- The default value or fixed value of an element or attribute, and enumeration values of a simple type, are now checked against the length, maximum length, and minimum length facets of the simple type, if supplied. If a value does not comply with the facets, an error appears in the Problems view of the message editor. Right-click the error and click **Quick Fix** to resolve the issue, if a quick fix is available. Otherwise, correct the problem manually using the editor. This error is most likely to occur if you import a COBOL copybook into the Version 5.0 Fix Pack 2 or earlier version of the Message Broker Toolkit.

- The CWF and TDS Encoding Null and Encoding Null Value properties have been removed from local attributes, global attributes, and attribute references. In XML Schema, only elements can have a Nillable property, therefore a null value cannot be carried by a data field that is modeled as an attribute. If you had specified CWF or TDS Encoding Null values for attributes in Version 5.0, they were ignored.

- The TDS Repeating Element Delimiter property has been removed from local attributes and attribute references. In XML Schema, attributes cannot repeat. If you had specified TDS Repeating Element Delimiter values for attributes in Version 5.0, they were ignored.

- For an element or attribute with simple type xsd:gMonth, the default value for the CWF, XML, and TDS Datetime Format property is now "--MM". In Version 5.0, the default value is "--MM--". This value has been corrected to comply with an XML Schema errata.

- When you are migrating, you do not have to redeploy your message sets because the mqsimigratecomponents command upgrades message dictionaries in the broker database to Version 6.1 format. However, if you do not redeploy a message set, and you observe a ParserException, such as a "checked vector error" or an "access violation", when you first use a message flow that uses the message set, you must redeploy the message set. This situation can occur if the message dictionary contains previously undetected bad data. In Version 6.1, a message dictionary is checked on first use; in Version 5.0, it is not.

Equivalent z/OS customization variables in mqsicompcif and BIPEDIT

If you are migrating from Version 5.0 to Version 6.1 on z/OS, customize the Version 6.1 BIPEDIT file to match the 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 5.0 mqsicompcif file, and the equivalent variables in the Version 6.1 BIPEDIT file so that you can set up the same values.

You can also look at the sample BIPEDIT file.

Version 5.0 mqsicompcif customization variables	Version 6.1 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
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

Version 5.0 mqsicompcif customization variables	Version 6.1 BIPEDIT customization variables
_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++

Part 3. Appendixes

103

Appendix. Notices for WebSphere Message 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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information includes examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

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

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

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

Trademarks in the WebSphere Message Broker information center

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ($^{®}$ or $^{™}$), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information at `www.ibm.com/legal/copytrade.shtml`.

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 of Intel Corporation in the United States and other countries.

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

Microsoft and Windows 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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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 5.0   40
      Version 6.0   16
BIPEDIT, equivalent variables in
  mqsicompcif   100
brokers
   migrating
      Version 5.0   50
      Version 6.0   23

C

coexistence   4
Configuration Manager
   migrating
      Version 5.0   44
      Version 6.0   20

D

data definitions, migrating   82
Database Instance Manager
   migrating from Version 6.0   33

H

HTTPRequest node, migrating   85

I

IA9O, removing   64

M

Message Broker Toolkit
   migrating
      from Version 5.0   42
      from Version 6.0   19
message flows
   migration notes   74
message sets
   migration notes   93
migration   3
   64-bit platforms   11
      user exits   11
      user-defined extensions   11
   brokers
      Version 5.0   50
      Version 6.0   23
   conditions
      Message Brokers Toolkit   6
      Version 5.0 brokers   8
   Configuration Manager
      Version 5.0   44
      Version 6.0   20
   data definitions   82

migration *(continued)*
   database tables, removing   62
   DatabaseInstanceMgr
      Version 6.0   33
   HTTPRequest node   85
   Message Broker Toolkit
      Version 5.0   42
      Version 6.0   19
   message flows, notes   74
   message mappings   75
      restrictions   76
   message sets, notes   93
   MQe flows   86
   ODBC, changing connection
     definitions
      Version 5.0   54
      Version 6.0   27
   planning
      Version 5.0   35
      Version 6.0   9
   removing IA9O   64
   removing obsolete message
     checks   63
   style sheets   81
   supported migration paths   73
   SupportPac IA9Q   17
   Toolkit
      Version 5.0   42
      Version 6.0   19
   unwanted files, removing   62
   User Name Server
      Version 5.0   59
      Version 6.0   31
   user-defined nodes   79
   Version 5.0 domain   35
      high availability   39
      large   38
      multiple   38
      small   37
   Version 6.0 domain   9
      high availability   15
      large   14
      multiple   14
      small   13
   XA resource manager definitions
      changing from Version 5.0   54
      changing from Version 6.0   27
   XML files   81
   XMLTransformation node   85
MQe connections, designing   88
MQe flows
   configuring after deployment   91
   deploying   89
   migrating   86
mqsicompcif, equivalent variables in
  BIPEDIT   100

O

obsolete messages   63

ODBC

ODBC
   connection
      changing from Version 5.0   54
      changing from Version 6.0   27

P

planning
   migration
      Version 5.0   35
      Version 6.0   9

R

restoring to previous versions   65

S

style sheets, migrating   81

T

Toolkit
   migrating
      Version 5.0   42
      Version 6.0   19
trademarks   107

U

upgrading
   supported upgrade paths   73
User Name Server
   migrating
      Version 5.0   59
      Version 6.0   31
user-defined nodes
   migrating   79

X

XML files, migrating   81
XMLTransformation node, migrating   85

IBM ®

Printed in USA