



IBM Software Group

WebSphere® Message Broker Version 6

Migration and coexistence



@business on demand.

© 2007 IBM Corporation
Updated March 28, 2007

This presentation discusses migration from earlier versions of WebSphere Message Broker, and how WebSphere Message Broker Version 6 can coexist with earlier broker versions.

Agenda

- Migration planning
- Migration steps
 - ▶ Migrating the configuration manager
 - ▶ Migrating the broker and user name server
 - ▶ Migrating application components
- Special considerations
- Summary and references



This presentation discusses migration planning and outlines the steps for migration of the various components.

Supported migration paths

- These products can be migrated to WebSphere Message Broker V6:
 - ▶ WebSphere MQ Publish/Subscribe Broker *
 - ▶ WebSphere Business Integration Event Broker V5 *
 - ▶ WebSphere Business Integration Message Broker V5
 - ▶ WebSphere MQ Integrator V2.1
 - ▶ WebSphere MQ Integrator Broker V2.1
 - ▶ WebSphere MQ Event Broker V2.1 *
- Recent Fix Packs required
- * **Highlighted** products can also be migrated to WebSphere Event Broker V6
 - You cannot remove function when migrating to V6
 - Unless otherwise stated, migration steps for WebSphere Event Broker are identical to the migration steps for WebSphere Message Broker



WebSphere Message Broker Version 6 supports migrations from a number of earlier versions. For this reason, the migration process has been improved significantly in Version 6, and has focused on reducing the specific order in which migration tasks need to be done.

All versions of message and event brokers at Version 2.1 or later can be migrated to WebSphere Message Broker Version 6. Those products highlighted in blue can also be migrated to WebSphere Event Broker Version 6. Note that some of these earlier versions can only be migrated if they are at a certain fix pack level. This varies depending on which version is being migrated; check the migration prerequisite statements in the Version 6 Information Center.

Earlier versions of message broker, for example MQSeries Integrator Version 2.0.2, cannot be migrated to Version 6 directly; it must be migrated to Version 2.1 as an interim step.

Features that will help you to migrate

- V6 can be installed on the same machine as, and run alongside, all previous versions of the product
- Domain coexistence
 - Complete interoperability between V5 and V6 components
 - V6 tools can manage V2.1 brokers
- Single new command to migrate components between versions
- No redeploys required, even from V2.1
- The V6 broker can be migrated back to V5
 - And to V2.1, requires broker Version 6, fix pack 3

5

Migration and coexistence

© 2007 IBM Corporation

The most significant change for WebSphere Message Broker Version 6 is that it can be installed alongside Version 5 or Version 2.1 of the broker. You do not need additional hardware to run concurrent Version 6 and V5 or V2.1 systems.

If you have a number of broker instances in a broker Domain, each of these brokers can be migrated individually to Version 6, while maintaining full interoperability between the instances. And, they can all be managed from a broker Version 6 toolkit.

When migrating to Version 6, a new command enables broker components to be migrated easily, and this means that broker artifacts such as flows or message sets will not need to be redeployed. This applies when migrating from either Version 5, or Version 2.1

In the event of a failure of the migration process, applications can be migrated back to the previous version. This is provided for both Version 5 and Version 2.1, although for Version 2.1, this requires Fix Pack 3 of WebSphere Message Broker Version 6.

Finally, note that applications must be redeployed if you are migrating a Configuration Manager to another machine.

Prerequisites

- V6 sees a significant reduction in the number of prerequisite products
- On Windows[®], the only prerequisites for unit test, development, proof-of-concept, and so on are:
 - ▶ WebSphere MQ (V5.3.0.1 or later)
 - ▶ ODBC access to Derby database (included with product)
- Broker database required for production and non-Windows
- Rational[®] Agent Controller required to use the debugger

First, the prerequisites to install a Version 6 broker environment are significantly simpler than either Version 5 or Version 2.1.

For a development and test system, the only prerequisite product is WebSphere MQ. This should be WebSphere MQ Version 5.3.0.1 or later, or Version 6.

Derby database support is embedded in the broker component on Windows only. You can use the embedded Derby database for verification, evaluation, and test purposes only. (Derby refers to IBM Cloudscape™ Version 10.0, a database product that is built on the Derby database from the Apache Software Foundation.) For a production system, install and configure your chosen enterprise database.

WebSphere Message Broker uses ODBC to connect to Derby databases. Derby is a native Java™ database engine without ODBC support. The DB2[®] Run-Time Client provides the drivers that allow ODBC to access Derby databases. The DB2 Run-Time Client is used only for providing and managing the ODBC connection between WebSphere Message Broker and the Derby databases. It does not provide a DB2 database and therefore does not consume the resources that a full DB2 installation normally requires.

To use the debugger, Rational Agent Controller is required.

Installing V6 alongside previous versions

- V6 product is installed to a new location
- Use profiles (“Command Console” on Windows) to select which instance you are working with
- Issue a command to migrate runtime components from one installation to another
- Component names must still be unique on the machine
 - ▶ It is now possible to name Configuration Managers, and have more than one running in each instance
- The V6 *mqsolist* command can list all components on the machine, or just those in the current instance



When WebSphere Message Broker Version 6 is installed, it is installed in a separate location from any earlier installed version. This is significant, because in earlier versions, the installation process automatically included the installation directory in the path statements. These were used on Windows for example, to set up the correct paths for broker Commands. In Version 6, since it is possible to have more than one version installed concurrently, broker Version 6 has introduced a new “Command Console”. This is available from the “Start, Programs” menu, or can be invoked as an icon from the Windows desktop. This Command Console should be used to issue broker Version 6 line commands.

If you have V5 broker installed on the same system, and you open a Command Prompt window to issue broker calls, then these commands will be directed to the Version 5 installation.

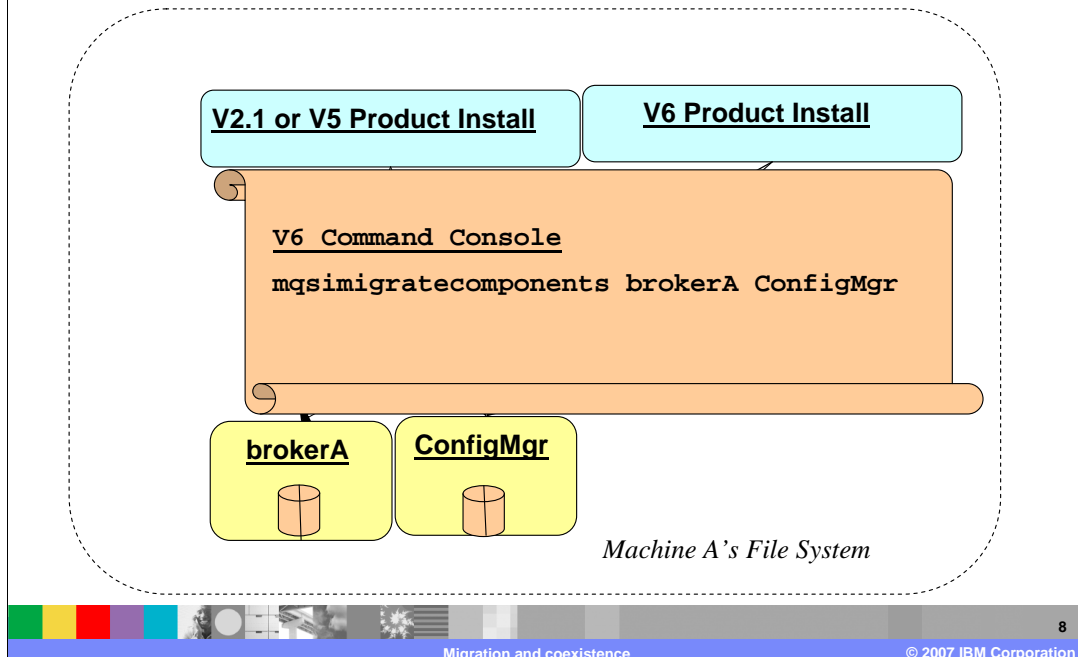
Component names must still be unique within the machine that you are running on. For example, if you have a broker B1 that was created with V5, then you cannot create another broker on the same system called B1 in V6.

This also applies to the Configuration Manager. In WebSphere Message Broker Version 6, it is possible to give the Configuration Manager a specific name. This also means that it is possible to have more than one Configuration Manager on one system. For compatibility with previous versions, it is possible to create an unnamed Configuration Manager on the Windows platform. However, if you create a Configuration Manager on any other platform, it must be named.

To check on which components have been created on a specific machine, use the *mqsolist* command to list all broker components for all versions.

If you plan to use the WebSphere Message Broker toolkit or the broker line commands on Linux[®], then the same process applies. However, when you create a Configuration Manager, it will have to have a specific name.

Installing V6 alongside previous versions

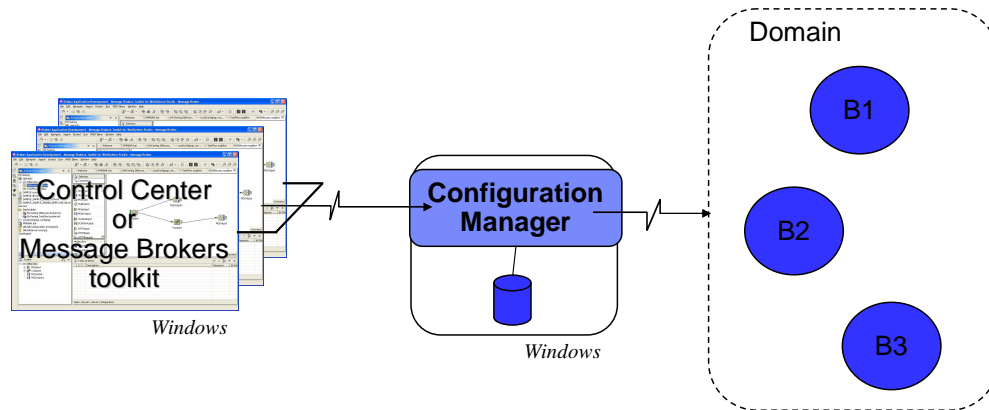


This slide shows at a high level how the migration process should be planned. Initially, broker V2.1 or V5 will be installed, and will have a configured broker and Configuration Manager. Assume that all components are installed on the same processor, so initially there is a broker called “brokerA”, and a Configuration Manager named “ConfigMgr”.

The first stage of the migration is to install WebSphere Message Broker Version 6 alongside the existing installation. From a Version 6 Command Console, issue the `mqsimigratecomponents` command, as shown on the slide. This command can take several parameters, which represent the components to be migrated. The example shows “brokerA”, and “ConfigMgr”. This will associate these components with the broker Version 6 installation libraries.

Since the broker V6 Configuration Manager does not need an external database, the database associated with the V2.1 or V5 Configuration Manager can be discarded.

A typical domain



9

This slide shows a typical broker configuration with the broker domain containing a number of broker instances.

V5 to V6 domain migration strategy

- Play with the technology
 - ▶ Developer installs V6 toolkit, Configuration Manager and broker alongside previous version on own machine. Creates V6 components and tries them out.
 - ▶ Migrate a broker
 - Migrate a test broker to V6. Ensure that existing flows run as expected.
 - ▶ Migrate the Configuration Manager
 - Could also move the Configuration Manager to a UNIX® or z/OS® platform at this point
 - ▶ Migrate more brokers
 - ▶ Migrate toolkits

10

Migration and coexistence

© 2007 IBM Corporation

Before actually performing a migration, it is most important to become familiar with the base components of Version 6. Some of the key system components have changed their nature, so using Version 6, and getting used to the new tools will make migration easier.

It is recommended that a new user install Version 6 locally, and use it to create some new Version 6 applications.

Next, perform a test migration of a Version 2.1 or Version 5 broker. run your message flows, using your usual testing tools.

Migrate a Configuration Manager. This can either be migrated to a Version 6 Configuration Manager on Windows, or to another platform. If you decide to change the platform for the Configuration Manager, this is likely to be moved to the same platform as the broker runtime.

Finally, the toolkit or Control center can be migrated to Version 6. All applications that were developed with the Version 5 or Version 2.1 toolkit can be imported into the Version 6 toolkit, with one exception. The Mapping node has changed in Version 6 and requires a special migrate command.

Section

Migration steps

This section covers migration steps.

How to migrate

- Prerequisites
- Components
 - ▶ Toolkit
 - ▶ Configuration manager
 - ▶ Broker
 - ▶ User name server
- Application logic
 - ▶ Message flows
 - ▶ Message sets
 - ▶ User defined nodes
 - ▶ Rules and formatters extension

The next slides discuss these topics in detail, and cover the base prerequisites for installation of Version 6, the migration of the core broker components, and the migration of different types of application component.

Domain coexistence and migration order

<i>Domain Compatible?</i>	V6 tools	V6 Configuration manager	V6 broker
V2.1 tools	No	No	No
V2.1 Config	No	n/a	No
V2.1 broker	Yes	Yes	Yes
V5 tools	Yes	Yes	Yes
V5 configuration manager	Yes	n/a	Yes
V5 broker	Yes	Yes	Yes

▪ **Migrating from V2.1 to V6**

Because the V2.1 Tools and Configuration manager do not work with V6 components, the migration order from V2.1 to V6 is:

- ▶ 1) Configuration manager and tools
- ▶ 2) brokers

▪ **Migrating from V5 to V6**

Because V5 components can completely interoperate with V6 components, there is no required migration order for V5 to V6.

- ▶ All toolkits that share a workspace must be migrated together

13

Migration and coexistence

© 2007 IBM Corporation

This slide shows domain compatibility and the order in which you should migrate components of the broker environment.

A Version 2.1 Control Center and a Version 2.1 Configuration Manager cannot connect to a Version 6 broker. In this case you should migrate the 2.1 Control Center and Configuration Manager first. When these have been migrated to Version 6, they will still be able to connect to, and manage, the Version 2.1 brokers.

When migrating Version 5 to Version 6, there is no requirement to migrate components in a particular order. All Version 5 and Version 6 components can interoperate, so the order of migration can be made based on your own requirements.

The table on this slide shows the cross-compatibility between Version 6 and earlier versions.

Migrating the toolkit

From V2.1

- Export your user data using the V2.1 Control Center
- Install the V6 toolkit (in a different path from the V2.1 Control Center)
- Import your user data into Eclipse projects
 - ▶ `mqsimigratemsgflows`
 - ▶ `mqsimigratemsgsets`
- Uninstall the V2.1 Control Center when no longer needed

From V5

- Install the V6 toolkit
- Backup your workspace
- Migrate any V5 mappings
 - ▶ `mqsimigratemfmaps`
- Create a new workspace and import your V5 data, or connect directly to your V5 workspace
- V5 message flows are automatically converted to a V6 format when you first save them. This format cannot be read by the V5 toolkit!

When migrating from a Version 2.1 Control Center, follow these steps:

- 1) Export user data
- 2) Install the V6 toolkit in a path different from the V2.1 Control Center
- 3) Import user data using the `mqsimigratemsgflows` and `mqsimigratemsgsets` commands
- 4) Uninstall the V2.1 Control Center when migration is complete and successful and the V2.1 Control Center is no longer needed

Note that it is a good practice to back up all application artifacts before starting any migrations.

The two commands shown on this slide to migrate Version 2.1 flows and message sets have been updated to handle migration to Version 6.

If migrating from Version 5, then the process is simpler. After installing the Version 6 toolkit, and creating a new workspace, import the Version 5 projects into the Version 6 workspace. If you import as an "Existing Project", you should not overwrite the ".project" file when you import into Version 6. Doing so could result in unresolved references in the Message Flow.

An alternative approach is to connect the Version 6 toolkit directly to the Version 5 workspace. If you do this, it is recommended to take a backup of the entire Version 5 workspace, before connecting it to the Version 6 toolkit. The first time that a Version 5 artifact is Saved with the Version 6 toolkit, it is changed into a Version 6 format. Therefore, do not attempt to reconnect the Version 5 toolkit to the workspace, once changes have been made in the Version 6 toolkit.

If you have used the Mapping node in Version 5, then you will need an additional step to convert the Version 5 maps to Version 6. This is done by using the `mqsimigratemfmaps` command.

The *mqsigratecomponents* command

- Migrates the data associated with a runtime component from one version to another, and assigns it to another instance
 - ▶ Migrates queues, databases and registry keys and files (where appropriate)
 - ▶ Can use the -1 -2 -3 flags to migrate each data type separately
 - ▶ Pre- and post-migration check options
 - ▶ Auto-recovery, back-out after failures

- Default operation is to move the specified components to the latest formats and data level
 - ▶ Use the -t <version> flag to back out V6 (broker only)

The *mqsigratecomponents* command migrates a number of broker components. If you use this command to run all components, it will attempt to migrate the registry settings, and any WebSphere MQ and database definitions. However, it is possible that a single person may not have the necessary authority to make changes to all of these resources, so the command can be supplemented with flag, which will indicate which resource is to be migrated.

To migrate the registry, use the flag “-1”; to migrate the WebSphere MQ definitions, use the flag “-2”; and to migrate database definitions, use the “-3” flag.

The “-t” option indicates “target”; this is used if you need to back out the migration, and the “-t” option indicates to which version the migration should be backed out.

Section

Migrating the configuration manager

This section covers migrating the configuration manager

Configuration manager migration considerations

- Configuration Manager is now available on Linux, AIX®, HP-UX, Solaris, Linux/390 and z/OS ... and Windows
- External databases are no longer used
 - ▶ Data is stored in an internal format
 - ▶ Transparent migration from DB2
 - ▶ Backup and restore provided
 - ▶ Cannot roll a changed domain back to DB2
- Multiple *named* Configuration Managers per machine
 - ▶ One per queue manager
- Ability to move Configuration Manager to another machine

17

Migration and coexistence

© 2007 IBM Corporation

The Configuration Manager is now available on all platforms supported by the broker runtime component. Also, in Version 6, the Configuration Manager does not use a DB2 database as its repository. Therefore, when migrating the Configuration Manager to Version 6, take these changes into account during the migration process.

The Version 6 Configuration Manager uses an internal repository; the format of this data is not published externally. However, the Configuration Manager repository can be backed up and restored using the commands *mqsibackupconfigmgr* and *mqsirestoreconfigmgr*.

It is important to note that if you migrate the Configuration Manager to Version 6, and then make changes, for example, create a new broker, you cannot backout the migration to Version 5. However, you can restore the original Version 5 Configuration Manager using a backup taken before the V6 migration.

Since it is possible to have multiple Configuration Managers on each system, the naming convention should be established before starting the migration process. If you plan to migrate the Version 5 Configuration Managers from Windows to another platform, the Configuration Manager must have a unique name. If you are migrating to Version 6 on Windows, then a unique name is not required, unless you plan to create more than one Configuration Manager. However, it is recommended that names are established for Configuration Managers.

Configuration manager migration scenarios

- In place
- To a new installation on a different machine, with DB2 installed
- To a new installation on a different machine, without DB2 installed

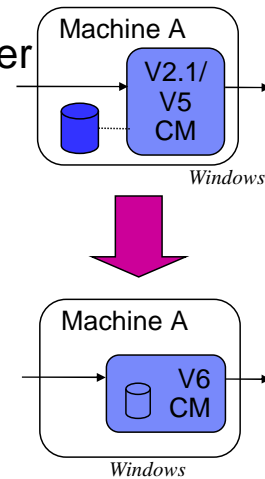
This presentation will explore three scenarios for migration of the Configuration Manager.

The first is a simple migration called “in place”, which means that the new Configuration Manager is on the same host machine as the old one. The old Configuration Manager could be at Version 2.1 or Version 5.

The second and third scenarios are where the Configuration Manager needs to be moved to a separate host system. The migration process is slightly different, depending on whether the target system has DB2 already installed or not.

Migrating the configuration manager in place

- Install V6
- Stop V2.1 or V5 Configuration Manager
- *mqsimigratecomponents*
- Start V6 Configuration Manager
- Uninstall (if needed)
 - ▶ DB2
 - ▶ Previous broker version



The steps for upgrading the Configuration Manager to Version 6 on the same system are

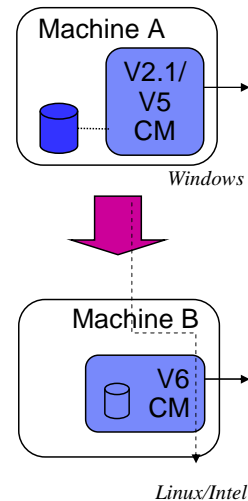
- 1) Install Version 6 alongside the existing installation
- 2) Ensure both the existing and new Configuration Manager components are stopped
- 3) run the *mqsimigratecomponents* command
- 4) Start the Version 6 Configuration Manager

The *mqsimigratecomponents* command only updates the appropriate registry settings, and does not actually migrate any data. The data migration happens when the new Version 6 Configuration Manager is started for the first time. At this point, the Configuration Manager recognizes that a migration is in progress, and copies all data from the old DB2 database to the new internal database. This could result in an extended startup time for the Configuration Manager, the first time this is started up.

Once the migration is working successfully, the old Configuration Manager can be uninstalled. Additionally, if there is no other requirement to have DB2 available, then this too can be uninstalled.

Migrating the configuration manager to another machine with DB2 installed

- Install V6 on new machine
- Create V6 CM, pointing it at existing CM's DB2 data source
- Stop V5 CM
- Start V6 CM (DB2 on V6 machine is used to migrate DB2 data automatically)
- Start deploy of the broker configuration to all brokers, so that the V6 CM can subscribe to broker status messages.
- Uninstall (if needed)
 - ▶ DB2
 - ▶ Previous broker version



20

Migration and coexistence

© 2007 IBM Corporation

To migrate a Configuration Manager to a new host system where DB2 is installed:

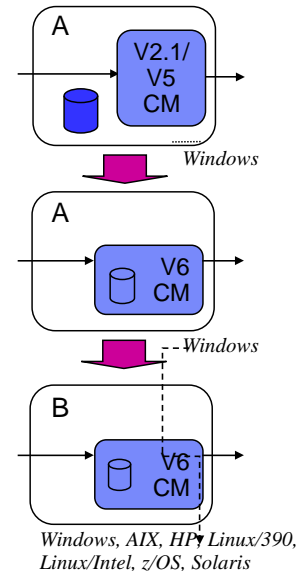
- 1) Install the Version 6 Configuration Manager on the target system
- 2) Create a Version 6 Configuration Manager, pointing it at the original DB2 database on the Version 5 or 2.1 system,
- 3) Stop the old Configuration Manager
- 4) Start the Version 6 Configuration Manager on the new system. This will have the effect of using the local DB2 on the new system to migrate the DB2 data in the old DB2 database. DB2 will be used as a route to connect to the old database, but the data will be loaded into the Configuration Manager internal repository.

Since all existing runtime broker instances will still be subscribed to the old Configuration Manager, refresh all of the existing brokers to update them with the new Configuration Manager details. This can be done by performing a broker configuration deployment to each of the brokers from the new Configuration Manager.

The Version 6 Configuration Manager is created by using the `mqsicreateconfigmgr` command. This is the same command as in earlier versions, except that the “-n” flag has changed its meaning. Because Version 6 does not have a DB2 repository, the “-n”, “-u”, “-p” flags serve a new purpose here, permitting migration from the old Configuration Manager repository during creation of the new Version 6 Configuration Manager. When using `mqsidedeploy` command, specify the broker to which you want to deploy using “b=” or “-b”, without indicating a bar file with the `-a` flag; this ensures that the broker configuration is deployed rather than a message flow application.

Migrating the configuration manager to another machine without using DB2

- Migrate Windows machine containing the V2.1 or V5 CM to V6
- Backup converted V6 data
- Install V6 on second machine and create CM
- Restore converted V6 data
- Start configuration deploy to all brokers, so that the V6 CM can subscribe to broker status messages.



21

Migrating a Configuration Manager to a target host where DB2 is not installed is a two stage process.

First, migrate the old Configuration Manager to Version 6 on the same host system. This requires an installation of Version 6 on the old host system. Backup the newly created Configuration Manager database by using the new `mqsibbackupconfigmgr` command. The output of this command is a zipped file, containing the new Version 6 Configuration Manager repository contents. Start the new Version 6 Configuration Manager at least once, before backing it up.

Second, install Version 6 on the target system, and create a new Configuration Manager. When creating the new Configuration Manager, do not specify a data source name. This will create a Configuration Manager with an empty repository. Then, using the zipped file created when you backed up the converted V6 data from the V6 Configuration Manager, run the `mqsirestoreconfigmgr` command against the V6 Configuration Manager on the target system. This will overwrite any data in the repository, so even if you create any new entries in the Configuration Manager, these will be overwritten when the restore command is run.

The target Configuration Manager can be any supported platform. The format of the zipped backup file is the same for all platforms, so this technique can be used to populate new Configuration Managers, including z/OS.

Section

Migrating the broker and user name server

This section discusses the details for migrating a broker and a User Name Server.

Migrating a broker considerations

- Derby (through DB2 Runtime Client) added as a new database option on Windows only
- Each part of the broker's configuration can be migrated separately
- Brokers can be migrated forwards and backwards
- Upgraded database drivers
 - ▶ `odbc.ini` changes documented in the Information Center

WebSphere Message Broker Version 6 is supported by the new internal Derby database. However, this is only available on Windows, and is only supported in a development and test system. It is not supported in a production environment.

Each part of the broker can be migrated separately. So, the registry, WebSphere MQ definitions and database configuration can all be migrated separately.

As part of migrating a broker, you must create ODBC definitions for broker and user databases, with database drivers that are appropriate for the new product version. You must do this before you run the `mqsimigratecomponents` command. See topic [ah23960_](#) in the Information Center for details on each platform.

Changes to *.odbc.ini* (Linux, UNIX platforms)

- V6 contains updated database drivers for Oracle and Sybase
- Need to change “UK***18.so” references to “UK***20.so”
- For example:

```
Driver=/usr/opt/wmqi/merant/lib/UKase18.so
```



```
Driver=/usr/opt/wmqi/merant/lib/UKase20.so
```

- Make the changes before migrating broker
- Users of XA co-ordination must make the same change to their qm.ini file

If you are using the broker on a UNIX platform, you will need to make changes to the ODBC driver for Version 6. This is done by making changes to the “odbc.ini” file.

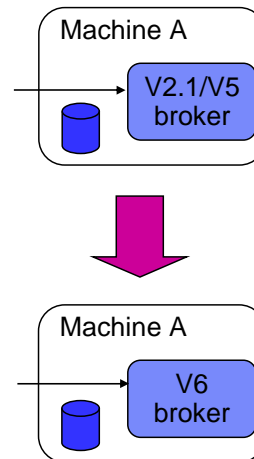
The main change that must be made is to change the “18” reference to “20”. For example, as shown on this slide, change the “18.so” to “20.so”, throughout the “odbc.ini” file. These changes must be made before migrating the broker, to ensure that the new broker can start correctly.

See topics ah23960 and ah14449 in the Information Center for details; search on the topic number in the Information Center.

If you use the WebSphere MQ Queue manager to perform XA transaction coordination between WebSphere MQ queues and the database, similar changes must be made to the “qm.ini” file. See topic ah13850 in the Information Center for details.

Migrating a broker (distributed platforms)

- Install V6
- Stop V2.1 or V5 broker
- *mqsigratecomponents*
- Start V6 broker
- Uninstall previous broker version (if needed)



To migrate a broker on all platforms except z/OS, perform the following steps:

First, install Version 6 broker alongside the existing Version 2.1 or 5 installation, and ensure that the old broker is stopped.

Then, run the *mqsigratecomponents* command. In this case, unlike the Configuration Manager case, the *mqsigratecomponents* command migrates the old broker database to the Version 6 database.

Finally, start the Version 6 broker. Starting the broker does not do any additional work the first time; the broker will be ready for work as soon as it successfully starts.

Migrating a broker (z/OS)

- Compared with V2.1 and V5, there are improvements to the way z/OS brokers are created and customized:
 - ▶ Create a new PDSE
 - ▶ Copy JCL from installed SBIPPROC/SBIPSAMP PDSEs and customize them
 - ▶ *mqsigratecomponents* run under the covers; use BIPMGxxx jobs
 - ▶ Copy new started task JCL to procedure's library
 - ▶ Verification program will run when the broker is started
- There are no scripts to run from the USS command line; everything is done from TSO.

Skeleton JCL is provided to migrate a Version 2.1 or Version 5 broker to Version 6 on z/OS. The specifics of setting up a broker under V6 are detailed in the z/OS Customization modules. A similar process would be followed to migrate a broker; skeleton JCL is provided to migrate a broker. After setting up the PDSE and performing the customization of the ODBC and environment files as outlined in the z/OS Customization modules, submit the BIPMGxxx jobs rather than the BIPCRxx jobs.

To migrate to V6 from V5, customize and submit the BIPMGCMP job. This migrates the registry, queues and broker database. As part of the database migration, database tables are created, or deleted, and dropped.

To migrate from V2.1 to V6, customize and submit the JCL BIPMGTB job. This creates LOB tablespaces. The broker database on z/OS introduced the use of LOB tablespaces in Version 5.0. They were not used in Version 2.1, so they need to be created at this point. Customize and submit the BIPMGCMP job. This migrates the registry, queues and broker database.

Once this has been done, ensure that the JCL of the broker started task is copied to the main procedure library.

In Version 6, all commands are now provided from the TSO command line. Although USS is used under the covers, there are no scripts that need to be run directly from USS. The BIPMGxxx jobs run the *mqsigratecomponents* command.

Migrating the user name server

- No functional changes since V2.1
- Migrate when appropriate
- Use *mqsigratecomponents* to move a V2.1 or V5 user name server to a V6 installation

If you have a User Name Server in your installation use *mqsigratecomponents* command to migrate after installing V6 alongside the existing installation.

Section

Migrating the application components

This section discusses migration of application components.

Migrating your application logic

- Message flows
- User defined extensions
- Message sets
- Broker archive files
- New-era-of-networks rules and formats

There are several different types of application logic that can be migrated: Message flows, user defined extensions, message sets, broker archive files, and “New Era of Networks” rules and formats.

WebSphere Message Broker Version 6 still supports applications written with the “New Era of Networks” tools, which can be deployed into the Version 6 broker.

Migrating message flows

- From V2.1

```
mqsimigratemsgflows -p <project> -d  
<directory>
```

- ▶ Iterates through <directory> looking for exported message flows
- ▶ Migrated flows are placed into Eclipse project <project>

- From V5

- ▶ V5 .msgflow files can be loaded into V6 workspace
- ▶ They are saved in a V6 format
- ▶ Therefore, V5 and V6 toolkits should not share a workspace



If migrating message flows from Version 2.1, the *mqsimigratemsgflows* command should be used to import projects into the Version 6 toolkit. Projects are exported from the Version 2.1 Control Center, and should be placed in a directory, which is used as input to this command.

If migrating from Version 5, the workspace which contains the Version 5 message flows can be accessed by the Version 6 toolkit. When the message flows are saved, they are saved in Version 6 format, which is different from Version 5 format. As a consequence, Version 5 and 6 Toolkits should not share a workspace, and once a workspace has been accessed by a Version 6 toolkit, it should not be accessed by a Version 5 toolkit.

Migrating user defined nodes

- From V2.1
 - *mqsimigratemsgflows* also discovers and migrates exported UDN definitions
- From V5
 - Import the user-defined node project into the V6 workbench and rebuild the project.
 - Modify the `<requires>` element in the `plugin.xml` file in the UDN project root to match the following:
 - `<requires>`
 - `<import match="greaterOrEqual" plugin="com.ibm.etools.mft.api" version="6.0.0"/>`
 - `</requires>`
 - Modify the "org.eclipse.help.contexts" extension in the same `plugin.xml` file to match the following:
 - `<extension point="org.eclipse.help.contexts">`
 - `<contexts file="HelpContexts.xml"/>`
 - `</extension>`
- From V5.1
 - Import the user-defined node project into the V6 workbench and rebuild the project.
- LIL file format has not changed since V2.1.

For migration from Version 2.1, the *mqsimigratemsgflows* command will automatically pick up User Defined node definitions, and migrate them. Hence, no further migration is necessary for this scenario.

When migrating from Version 5, follow the steps shown on the slide. The migration does require some hand-coded changes to the "plugin.xml" file in the User Defined node project; set these values to those shown on the slide to point to the Version 6 broker.

If you have the toolkit Version 5.1, then all that is required is to import the project into the Version 6 toolkit, and rebuild the project.

The format of the "LIL" file is unchanged from Version 2.1, so no changes will be necessary for this. However, the "LILPath" directory statements may need to be changed for Version 6.

Finally, if you have any user-written parsers, these can be treated as a LIL file. The format is unchanged, although the load path may need to be adjusted for the new installation.

Migrating message sets

- From V2.1

```
mqsimigratemsgsets -data <workspace> -d  
<directory>
```

- ▶ iterates through <directory> looking for exported message sets
- ▶ Migrated sets are placed into Eclipse workspace <workspace>
- ▶ One project is created per export file

- From V5

- ▶ Load V5 message set projects into the V6 workspace
- ▶ The format of message set files has not changed

The approach for Message Sets is very similar to the process for Message Flows.

When migrating from Version 2.1, use the *mqsimigratemsgsets* command, specifying the directory containing the exported message sets from Version 2.1, and the target Version 6 workspace.

When migrating from Version 5, message set projects can be accessed directly by Version 6, so the Version 6 toolkit can access the original workspace directly.

Migrating broker archive files

- The bar file format has not changed from V5
 - ▶ Both V5 and V6 bar files can be deployed:
 - using the V5 or V6 toolkit (or *mqsdeploy*)
 - through the V5 or V6 Configuration Manager
 - to V2.1, V5 and V6 brokers
- More file types can be deployed to V6 brokers
 - ▶ .jar, .xsl and .xml in addition to .cmf and .dictionary.
 - ▶ Must be deployed through a V6 Configuration Manager (otherwise, files of the the new types are ignored)
- The V6 Configuration Manager extracts version information from all bar files

The broker archive file format has not changed in Version 6, and is still based on the standard “.zip file” format. A bar file therefore, can be accessed by a Configuration Manager and broker at any version, and can be deployed from a Version 5 Configuration Manager to a Version 6 broker.

However, with the new function available in Version 6, a bar file that is generated by a Version 6 toolkit must be deployed using a Version 6 Configuration Manager into a Version 6 broker. So, if a bar file contained any of the new artifacts such as “jar”, “xsl”, or “xml” files, and it was deployed using a Version 5 Configuration Manager, these new artifacts would be ignored.

Finally, the Version 6 Configuration Manager contains new functionality to report the versions of the deployed components, such as Compiled Message Flows or Message Sets. If the bar file is deployed using a Version 5 Configuration Manager, this information will not be available.

Migrating rules and formats extension

- There are no changes to the rules and formats database
- The .FIE or .RIE data files used to hold rules and formats data have not changed
- No additional user actions over and above the migration needed for the message broker
- The Rules and Formatter component of the product is upgraded at the same time as the rest of the product

This slide discusses migration of applications built with the New Era of Networks Rules and Formatter tools.

This function has had no changes for Version 6, and the support is exactly the same as in Version 5. The internal database structure and the “FIE” and “RIE” data files is unchanged, and no specific actions are needed to migrate these applications over the normal migration approach.

Section

Special considerations

This section highlights some special considerations when migrating to WebSphere Message Broker V6.

Things to be aware of

- MQe nodes
- XMTransformation node
- Command console on Windows

You should take into consideration specific behavior if migrating MQe nodes and XMLTransformation nodes to WebSphere Message Broker V6. Also, note the behavior of line commands on Windows and use of the Command Console.

MQe nodes

- MQeInput and MQeOutput nodes have been deprecated in WebSphere Message Broker Version 6
- The runtime implements these nodes as MQInput and MQOutput nodes
- Because of this, you must configure your WebSphere MQ queue manager to communicate with the MQe queue manager; use the MQe Administration Wizards supplied with WebSphere MQ Everyplace® to do this.
- Flows using the MQeInput and MQeOutput nodes are not automatically migrated by the WebSphere Message Broker toolkit.

37

Migration and coexistence

© 2007 IBM Corporation

MQ Everyplace nodes have not been implemented in WebSphere Message Broker Version 6. WebSphere MQ Everyplace is an application designed primarily for messaging to, from, and between pervasive devices, such as mobile phones and PDAs. WebSphere MQ Everyplace supplies the MQe function.

The operation of WebSphere MQ Everyplace is different from that of WebSphere Message Broker, which means that there are different concepts involved in using a broker to operate with WebSphere MQ Everyplace. In particular, the message format is different, and messages must be converted.

Communication between WebSphere MQ Everyplace and a broker is achieved through the use of an external MQe Gateway, also called a *bridge-queue-manager*. WebSphere MQ Everyplace must be installed separately, and its MQe Gateway configured appropriately. Your flows use MQInput and MQOutput nodes to communicate with the gateway using MQ format messages. The gateway then communicates with MQe devices using MQe format messages.

If you migrate message flows which contain MQeInput or MQeOutput nodes to a Version 6 broker, the broker will treat these nodes as normal MQInput or MQOutput nodes. Consequently, you will need to make changes to the underlying queue manager and bridge to the MQe Gateway Queue Manager. See topic ac34700 in the Information Center for details on migrating MQe nodes.

XMLTransformation node

- Migration issues only if you reference stylesheets and XML files using a relative pathname
- In previous versions of the product, documents were loaded relative to the directory in which a broker started.
- In V6, they are loaded relative to the XSL/external and XML/external directories (for XSL and XML files) in the broker's data directory.
- You may need to change your stylesheets and XML files so that they can still be found after migration.
- Recommendation:
 - ▶ Change to deploy XML stylesheets in a bar file

38

Migration and coexistence

© 2007 IBM Corporation

The implementation of XML stylesheets has changed between broker Version 5 and Version 6. This facility was not available with Version 2.1

In Version 5, the XMLTransformation node referenced an XML stylesheet, and the location of this stylesheet was specified relative to the directory of the broker runtime code. The stylesheet had to be manually copied to the target directory, so that it could be located correctly by the broker.

In Version 6, the stylesheet location has been changed. Stylesheets should now be placed in the "XSL/external" or "XML/external" directories. This may mean that message flows which specify a stylesheet location will need to be changed to reflect this.

However, the recommended approach for deployment of XML stylesheets in Version 6 is to use the bar file mechanism. This approach will avoid any possible confusion over directory locations.

Command console on Windows

- Previous versions modified the system environment (PATH) so that you could type in all *mqs** commands from any command prompt
- V6 has a Command Console, in order to enable coexistence
- Available on the Start Menu or *[installdir]\bin\mqsiprofile.cmd*
- Remember to modify scripts so that the correct environment is set up!
 - `@call [installdir]\bin\mqsiprofile.cmd`

39

Migration and coexistence

© 2007 IBM Corporation

Since it is possible to run Version 6 alongside earlier versions, the Version 6 broker has introduced a Command Console on Windows; this is used to run commands against specific Version 6 components.

Earlier versions made modifications to the Windows system PATH, so that broker commands could be run from any Windows Command Prompt. Version 6 does not make this global change; you must issue Version 6 commands only from the Version 6 Command Console.

If you have written your own scripts to use the supplied broker commands, remember to make the appropriate changes to the PATH statements, so that the correct environment is used.

Section

Summary and references

Following is a summary of migration and coexistence of WebSphere Message Broker products.

Summary

- Simplifying migration has been a key focus area for WebSphere Message Broker Version 6
 - ▶ Coexistence of Version 6 with Version 5 or Version 2.1
 - ▶ Ability to mix and match toolkit, Configuration Manager and broker components at different versions
 - ▶ Migration back-out support

So to summarize, WebSphere Message Broker Version 6 provides the ability to have the new version installed alongside the older versions.

The ability to combine broker components at different levels is provided, and there are a number of tools to enable migration of broker components and user applications to WebSphere Message Broker Version 6.

There is now the ability to back out the migration, in the event of problems in the migration process.

References

- WebSphere Message Broker library:

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

- WebSphere Message Broker Information Center:

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

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

[Click to send e-mail feedback](#)

Trademarks, copyrights, and disclaimers

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

AIX DB2 Everyplace Rational WebSphere z/OS

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

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

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

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

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

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

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

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

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

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

© Copyright International Business Machines Corporation 2007. All rights reserved.

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

