IBM Software Group

WebSphere® Message Broker version 6.1

Migration and coexistence

business on demand.

Updated January 22, 2008

This presentation will give an overall view of how to migrate an existing Message Broker installation to WebSphere Message Broker Version 6.1.

Agenda

- Overview
- Co-existence
  - Product Installation co-existence
  - Technical co-existence
- Migration
  - Of developed artifacts
  - Of product components

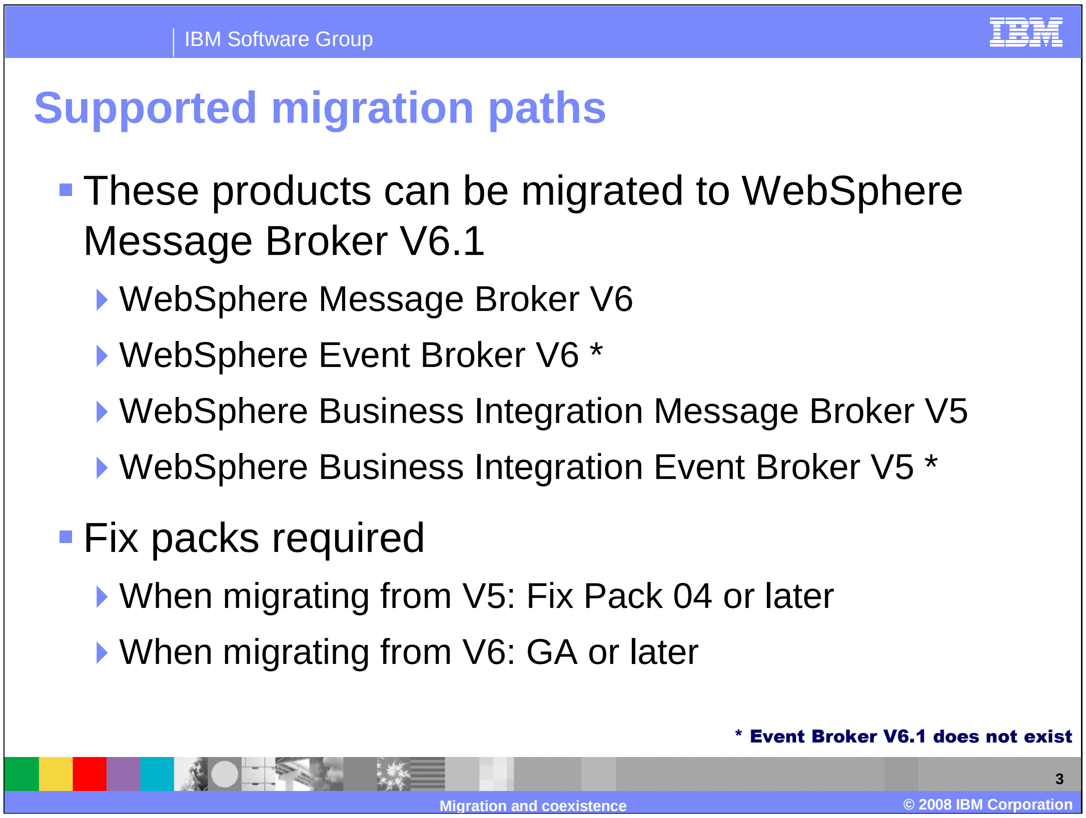
This presentation gives an overview of how to migrate to Message Broker version 6.1.

It will start with a discussion of the features in Message Broker version 6.1 that affect migration.

The presentation will then move on to talk about co-existence. Message Broker version 6.1 can be installed alongside previous versions, and can run the various product components concurrently with components from previous versions.

The final section, the majority of this presentation, deals with the migration of your message flows and any other Message Broker assets, and how to migrate the main components of the product.

This presentation does not cover the new capabilities of version 6.1; it is assumed that you are already aware of the benefits of Message Broker version 6.1 and have already made the decision to migrate.



Supported migration paths

- These products can be migrated to WebSphere Message Broker V6.1
  - WebSphere Message Broker V6
  - WebSphere Event Broker V6 *
  - WebSphere Business Integration Message Broker V5
  - WebSphere Business Integration Event Broker V5 *
- Fix packs required
  - When migrating from V5: Fix Pack 04 or later
  - When migrating from V6: GA or later

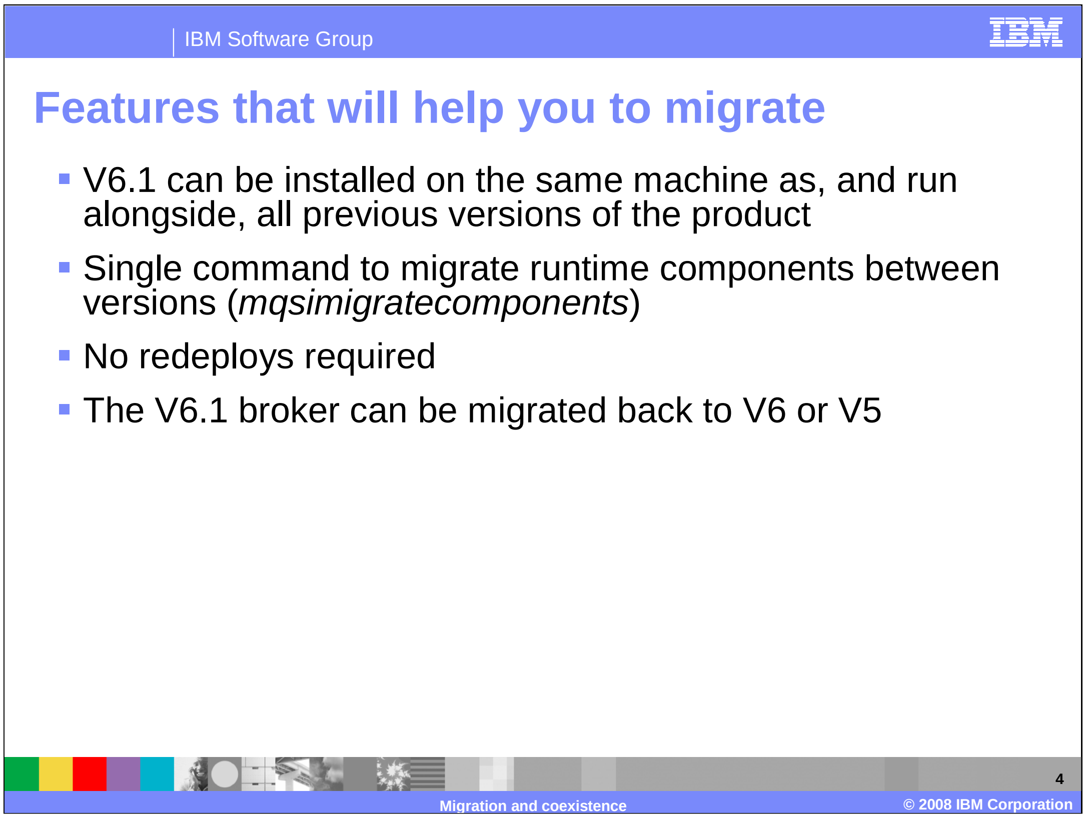
*** Event Broker V6.1 does not exist**

It is possible to migrate to WebSphere Message Broker version 6.1 from previous versions of the product; Migration from version 6.0 is supported at any fix pack level. Migration from version 5 is supported, but you are required to be using fix pack 4 or later.

Migration is also possible from version 5 and version 6.0 of Event Broker (at the same fix pack levels as described above). Event Broker does not exist as a separate product in version 6.1. All Event Broker capability exists within WebSphere Message Broker version 6.1.

Unless otherwise stated, the techniques used for migration is identical for all of the products described.

Finally, it is not possible to migrate from version 2.1 or earlier versions of the product. If you are migrating to version 6.1 from version 2.1, it is recommended that you first migrate to WebSphere Message Broker version 6.0.



IBM Software Group

Features that will help you to migrate

- V6.1 can be installed on the same machine as, and run alongside, all previous versions of the product
- Single command to migrate runtime components between versions (*mqsimigratecomponents*)
- No redeploys required
- The V6.1 broker can be migrated back to V6 or V5

4

Migration and coexistence

© 2008 IBM Corporation

Making migration as pain-free and straightforward as possible is an important aim for WebSphere Message Broker version 6.1, and there are several features in the product to help this.

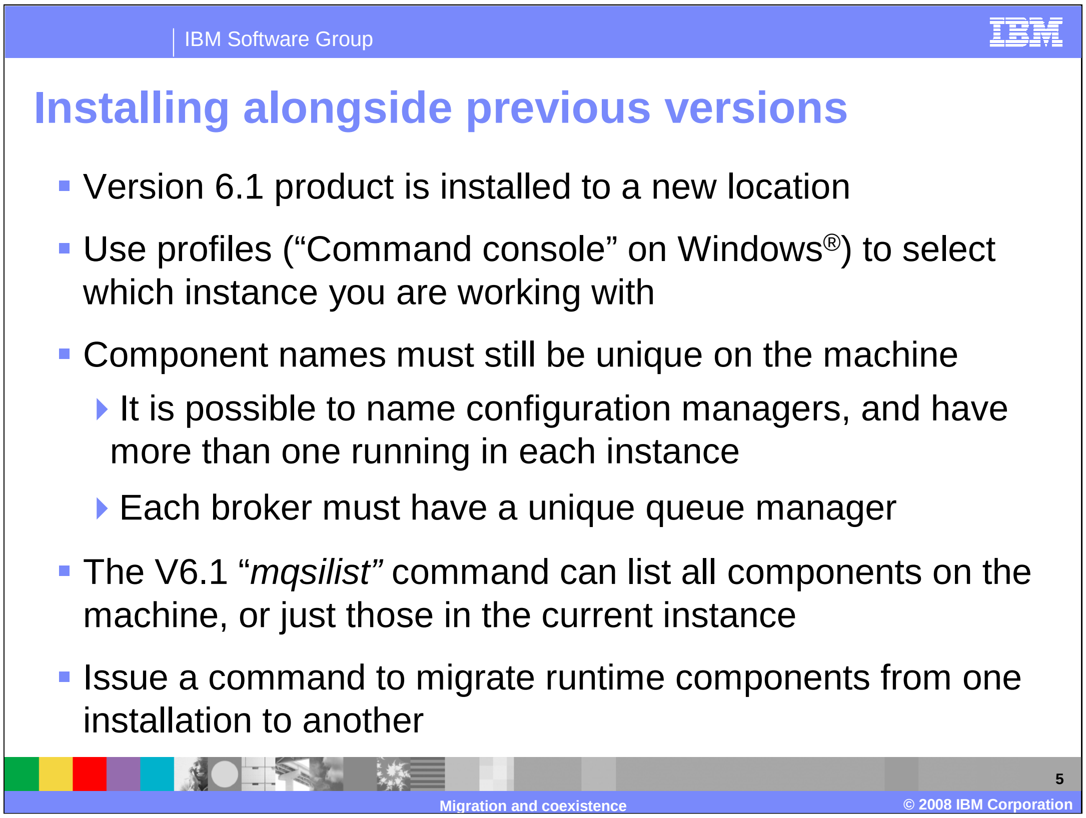
Firstly, on all platforms, WebSphere Message Broker version 6.1 can be installed and run alongside previous versions of the product. This means that your previous version can be running while you are setting up version 6.1. This removes the need for additional hardware when migrating a business-critical system.

Migration of the runtime components is achieved using a single command – “mqsi-migrate-components”. This migrates all of the broker’s internal definitions to the new version.

Once you have migrated your existing brokers to version 6.1, you don’t have to redeploy any of your development artifacts in order to make them work on version 6.1. Your brokers will start running your existing message flows as soon as you bring up the broker at the new version. This reduces the time and effort required to migrate a broker.

If you find that you need to revert to the previous version of Message Broker, the mqsi-migrate-components command is able to revert a migrated broker back to the previous version. This reduces the business risk associated with a failed migration. (Of course, this is not a substitute for taking adequate backups before attempting any migration.)

These topics will be covered in more detail on subsequent slides.



IBM Software Group

Installing alongside previous versions

- Version 6.1 product is installed to a new location
- Use profiles (“Command console” on Windows®) to select which instance you are working with
- Component names must still be unique on the machine
  - It is possible to name configuration managers, and have more than one running in each instance
  - Each broker must have a unique queue manager
- The V6.1 “*mqsilist”* command can list all components on the machine, or just those in the current instance
- Issue a command to migrate runtime components from one installation to another

5

Migration and coexistence

This describes in more detail how the multiple-installs concept works.

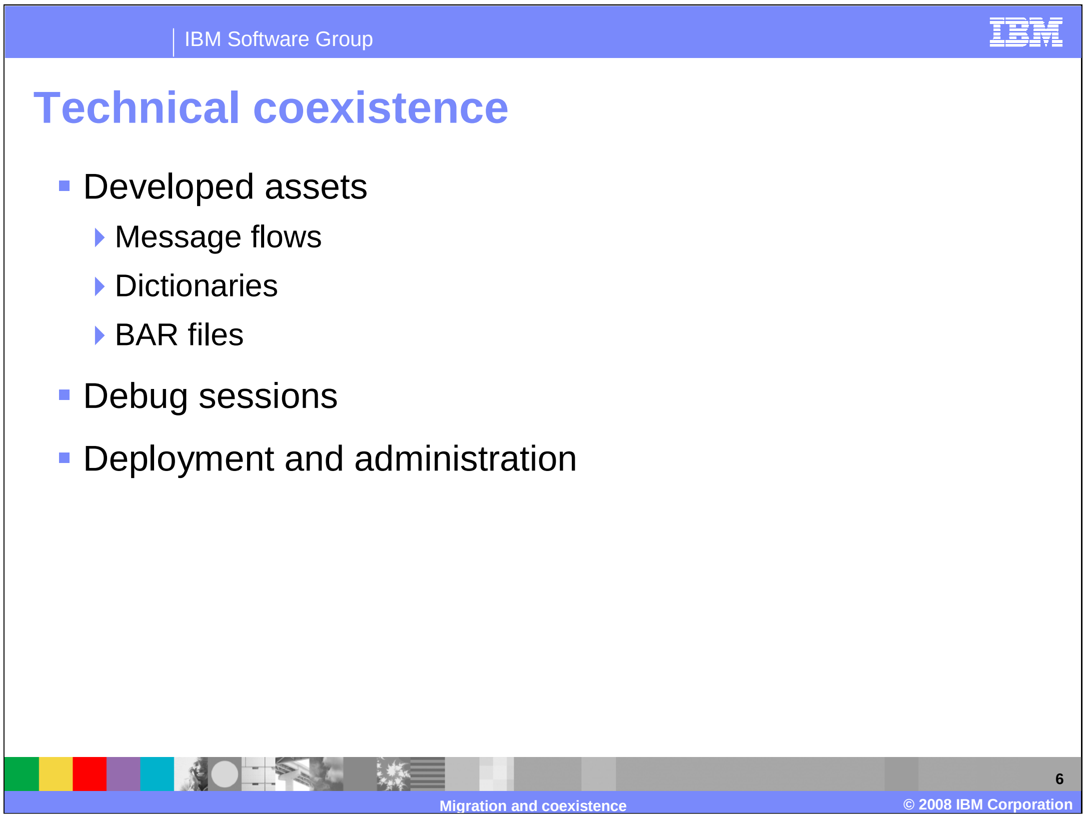
When you install WebSphere Message Broker version 6.1 on a machine that already has a previous version installed, the product code is placed in a completely new location on the file system. This means that while at least two versions of the product exist on the machine, you need to qualify any commands you enter, so that they are directed at the correct installation.

To do this, WebSphere Message Broker version 6.1 uses a concept called ‘profiles’. Commands are entered into a shell which has its environment configured for a specific version. So for example, commands for version 6 are entered in one window; commands for version 6.1 are entered in another. On Windows machines, this shell is called the ‘Command Console’ and is found on the Start Menu in the IBM WebSphere Message Broker version 6.1 category.

The only restriction on this capability is that component names must be unique on the machine. For example, you can’t have a version 6.0 broker and a version 6.1 broker, where both brokers have the same name. Nor is it possible for two brokers of the same or different versions to share queue managers. This helps to remove any confusion when coexisting versions.

With this in mind, both the 6.0 and 6.1 versions of the “mqsi-list” command are able to list brokers of any version. The –a flag will list all components on the machine. If the flag is not specified, only the components for the profile to which the command was directed are listed.

The mqsi-migrate-components command, which will be described in more detail shortly, is used to move components associated with one profile on a machine to another profile.



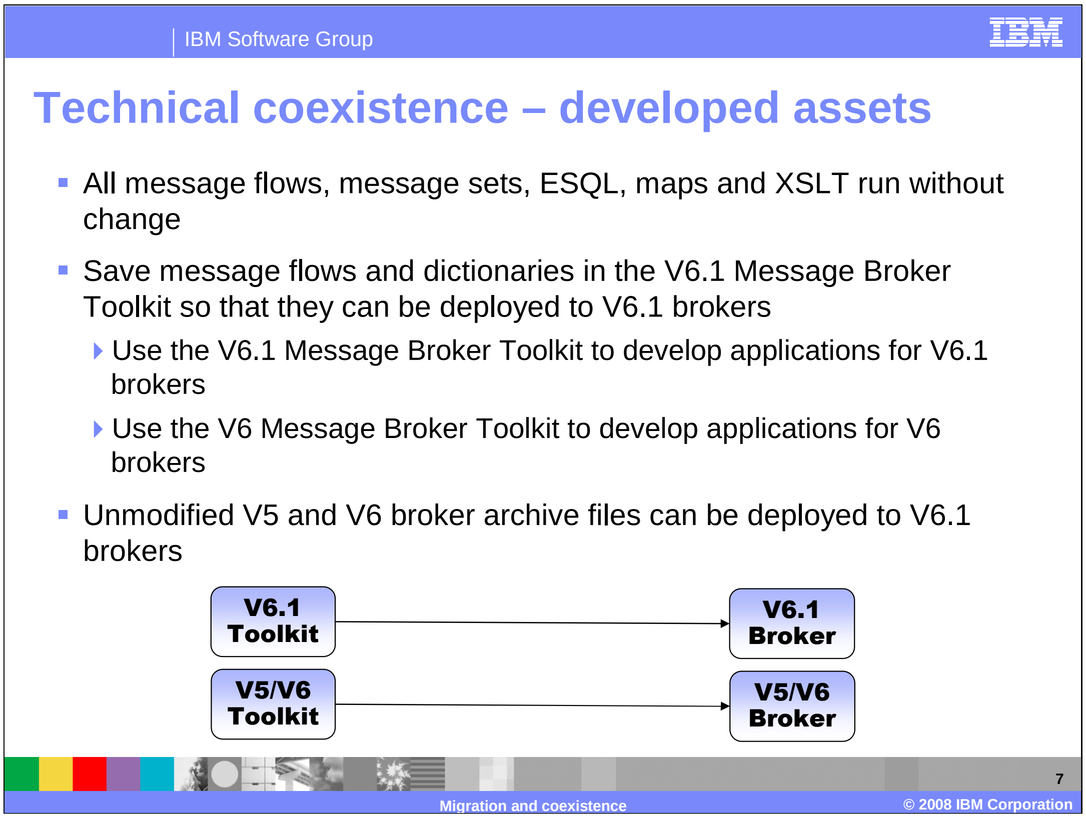
Technical coexistence

- Developed assets
  - Message flows
  - Dictionaries
  - BAR files
- Debug sessions
- Deployment and administration

Technical co-existence is the extent to which components from previous versions of the product will work with Message Broker version 6.1 components.

There are three aspects to this; first, the co-existence of developed assets, and which broker versions those artifacts can work with, second, the co-existence of toolkit and broker during a debug session, and third, the extent to which broker domains can be managed across versions.

The next few slides discuss these in more detail.



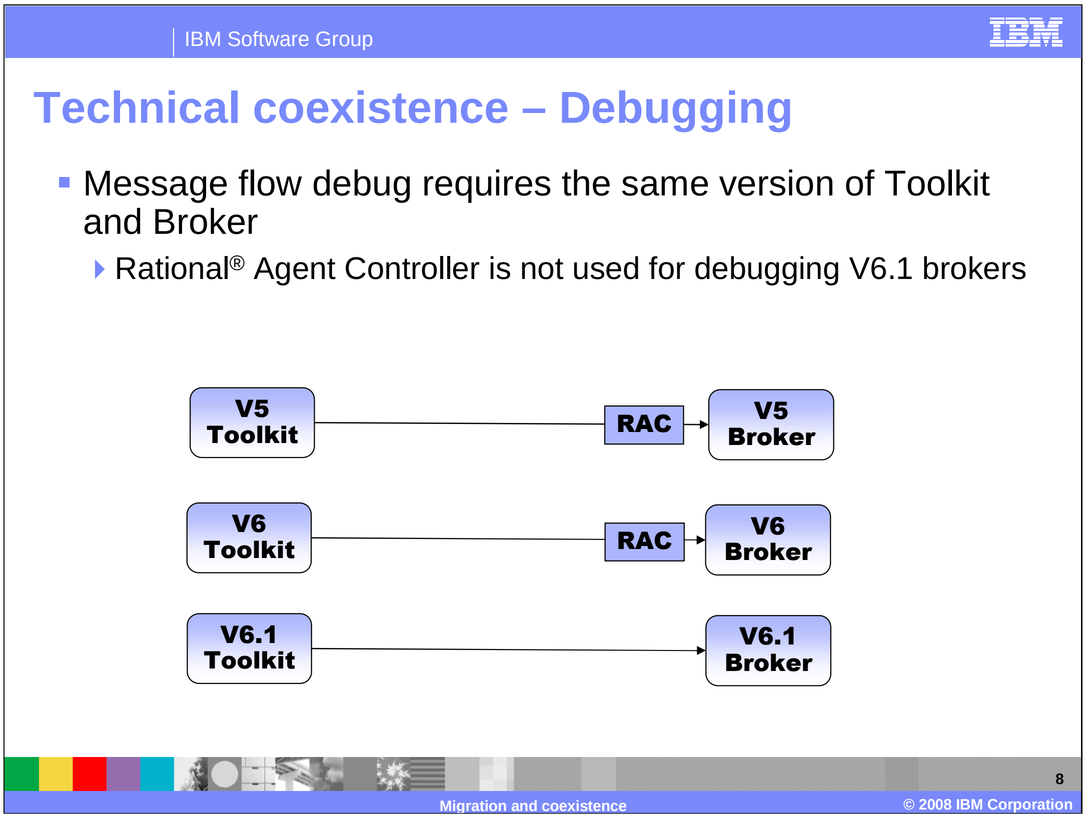
Technical coexistence – developed assets

- All message flows, message sets, ESQL, maps and XSLT run without change
- Save message flows and dictionaries in the V6.1 Message Broker Toolkit so that they can be deployed to V6.1 brokers
  - Use the V6.1 Message Broker Toolkit to develop applications for V6.1 brokers
  - Use the V6 Message Broker Toolkit to develop applications for V6 brokers
- Unmodified V5 and V6 broker archive files can be deployed to V6.1 brokers

You do not need to modify your developed assets in order to make them run with version 6.1. As soon as message flows and dictionaries are saved and built in the version 6.1 Message Broker Toolkit, they become artifacts that can be used with version 6.1 brokers.

This means that while you have version 5 and version 6 brokers to maintain, you need to have a version 5 or version 6 Message Broker Toolkit around in order to develop applications for them. As it is possible to install the version 6.1 Message Broker Toolkit alongside the earlier versions of the Message Broker Toolkit, you may need to switch between toolkits if you are working in a mixed development environment.

However, unchanged broker archive files can still be deployed to earlier brokers through the version 6.1 Message Broker Toolkit. It is also possible to perform administration of version 5, 6.0 and 6.1 brokers through the version 6.1 Message Broker Toolkit. This will be discussed later in this presentation.



Technical coexistence – Debugging

- Message flow debug requires the same version of Toolkit and Broker
  - Rational® Agent Controller is not used for debugging V6.1 brokers

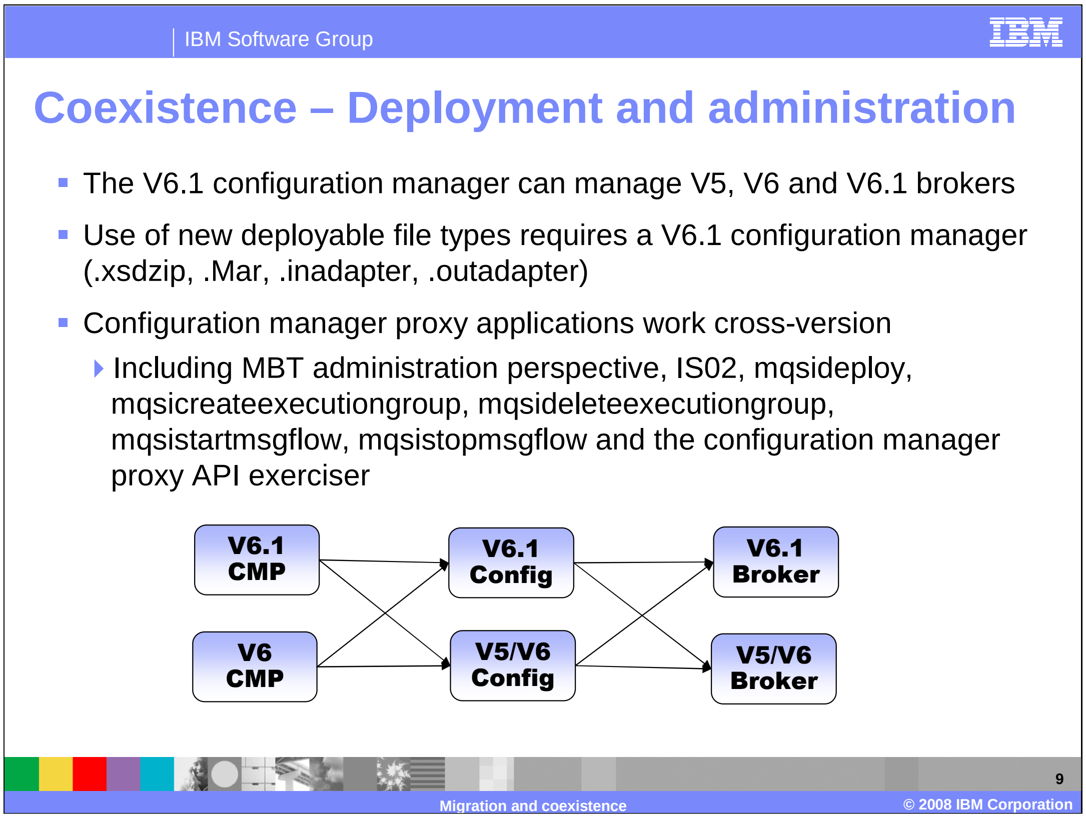
V5 Toolkit → RAC → V5 Broker

V6 Toolkit → RAC → V6 Broker

V6.1 Toolkit → V6.1 Broker

In version 6.1, Message Broker no longer uses the Rational Agent Controller for debugging message flows.

This means that the version 6.1 Message Broker Toolkit can only debug message flows that are deployed to a version 6.1 broker. The version 6.0 Message Broker Toolkit can only debug message flows that are deployed to a version 6.0 broker.



IBM Software Group

Coexistence – Deployment and administration

- The V6.1 configuration manager can manage V5, V6 and V6.1 brokers
- Use of new deployable file types requires a V6.1 configuration manager (.xsdzip, .Mar, .inadapter, .outadapter)
- Configuration manager proxy applications work cross-version
  - Including MBT administration perspective, IS02, mqsideploy, mqsicreateexecutiongroup, mqsideleteexecutiongroup, mqsistartmsgflow, mqsistopmsgflow and the configuration manager proxy API exerciser

9

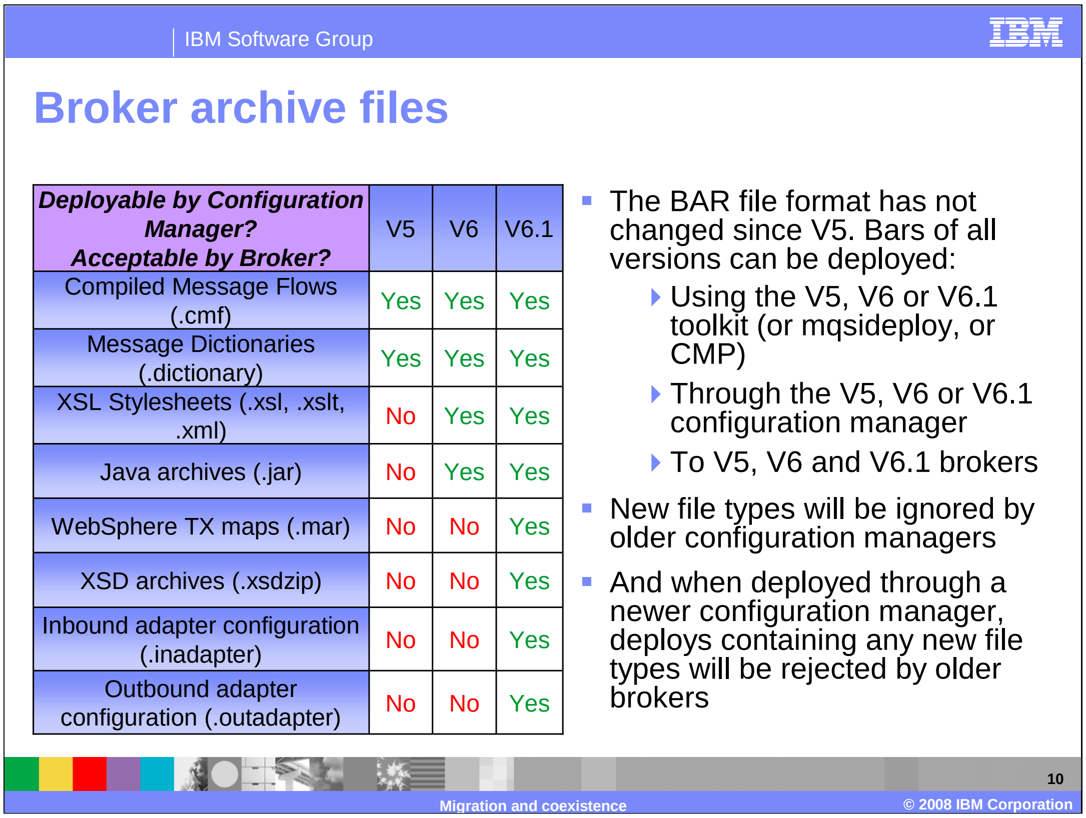
Migration and coexistence

© 2008 IBM Corporation

Broker archive files must be built using a Message Broker Toolkit of the same version as the broker to which they will be deployed. However, the deployment itself can be done using a configuration manager of an earlier version, unless file-types new to version 6.1 are used.

Additionally, the version 6.1 administration tools shown on this slide will work across different versions. This means that, for example, it is possible to connect to and manage domains controlled by a version 6.0 configuration manager from within the version 6.1 Toolkit Administration Perspective.

Any configuration manager Proxy applications you have developed in Message Broker version 6.0 will continue to work with the version 6.1 configuration manager Proxy classes. However, as the configuration manager Proxy is now compiled with Java 5, it is recommended that you recompile any applications that you have developed to run on Java 5. This will allow them to benefit from the improvements to type-checking that Java 5 provides.



IBM Software Group

Broker archive files

Deployable by Configuration Manager? Acceptable by Broker?	V5	V6	V6.1
Compiled Message Flows (.cmf)	Yes	Yes	Yes
Message Dictionaries (.dictionary)	Yes	Yes	Yes
XSL Stylesheets (.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 BAR file format has not changed since V5. Bars of all versions can be deployed:
  - Using the V5, V6 or V6.1 toolkit (or mqsideploy, or CMP)
  - Through the V5, V6 or V6.1 configuration manager
  - To V5, V6 and V6.1 brokers
- New file types will be ignored by older configuration managers
- And when deployed through a newer configuration manager, deploys containing any new file types will be rejected by older brokers

10

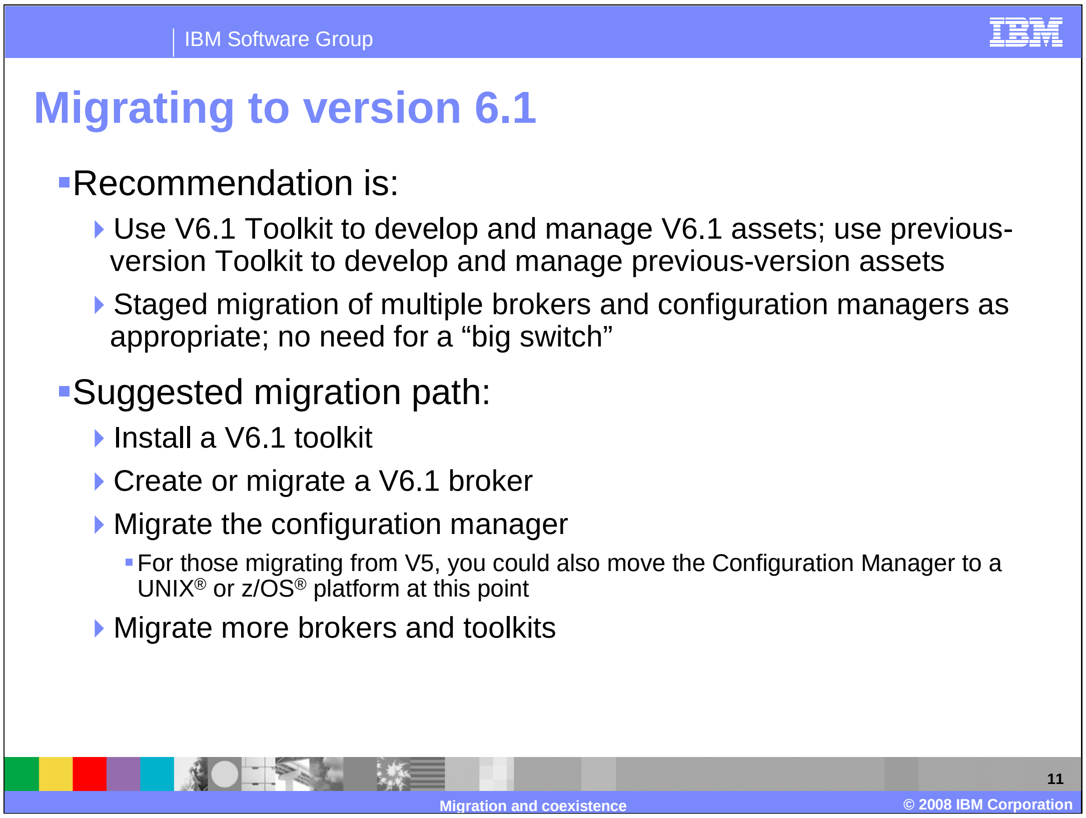
Migration and coexistence

This slide describes in more detail which file types can be deployed at the different versions.

The broker archive file format has been used by the broker since version 5.0, although the number of deployable files containable within has increased with each new version. Version 6.0 added support for XSL style-sheets and Java archives; version 6.1 adds support for WebSphere TX maps, XSD archives and adapter configurations.

The version 6.1 configuration manager can deploy all of these file-types. Earlier configuration managers will ignore file-types it doesn’t understand. If you continue to use a version 5 configuration manager, you will not be able to use the “runtime version” capability that was first added in message broker version 6.0.

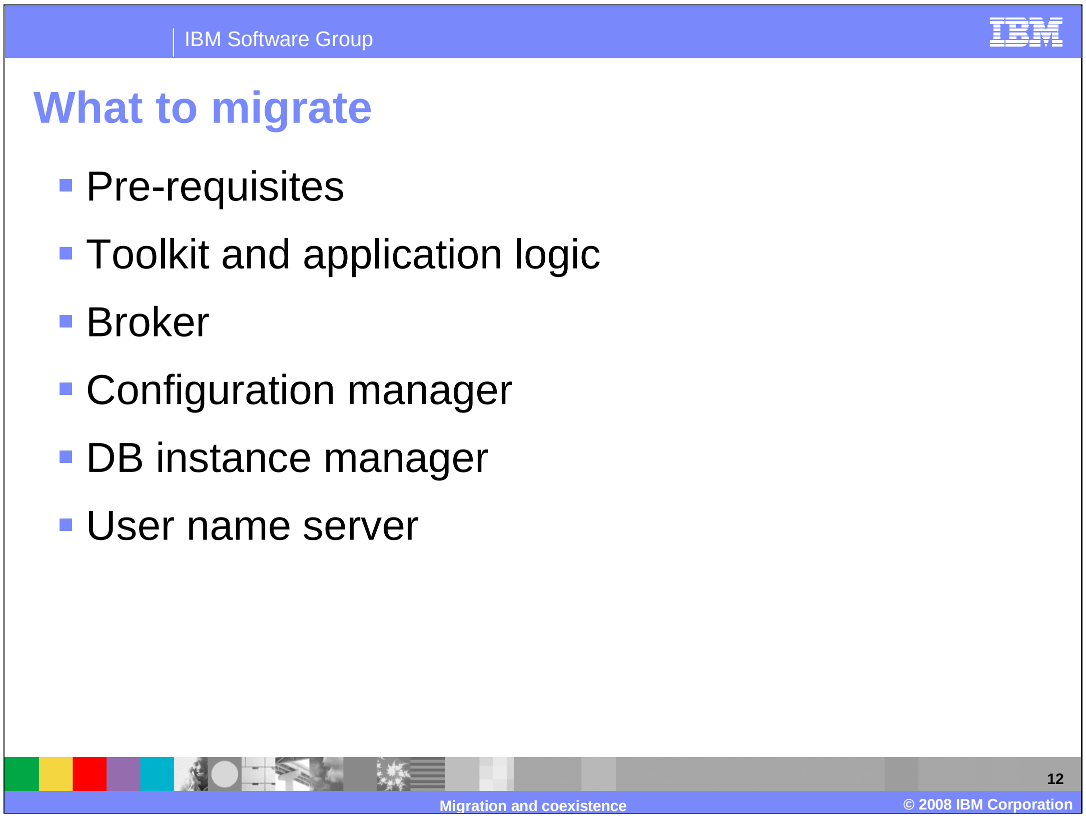
If you attempt to deploy newer file-types through to an older broker, the broker will reject the deployment.



Migrating to version 6.1

- Recommendation is:
  - Use V6.1 Toolkit to develop and manage V6.1 assets; use previous-version Toolkit to develop and manage previous-version assets
  - Staged migration of multiple brokers and configuration managers as appropriate; no need for a “big switch”
- Suggested migration path:
  - Install a V6.1 toolkit
  - Create or migrate a V6.1 broker
  - Migrate the configuration manager
    - For those migrating from V5, you could also move the Configuration Manager to a UNIX® or z/OS® platform at this point
  - Migrate more brokers and toolkits

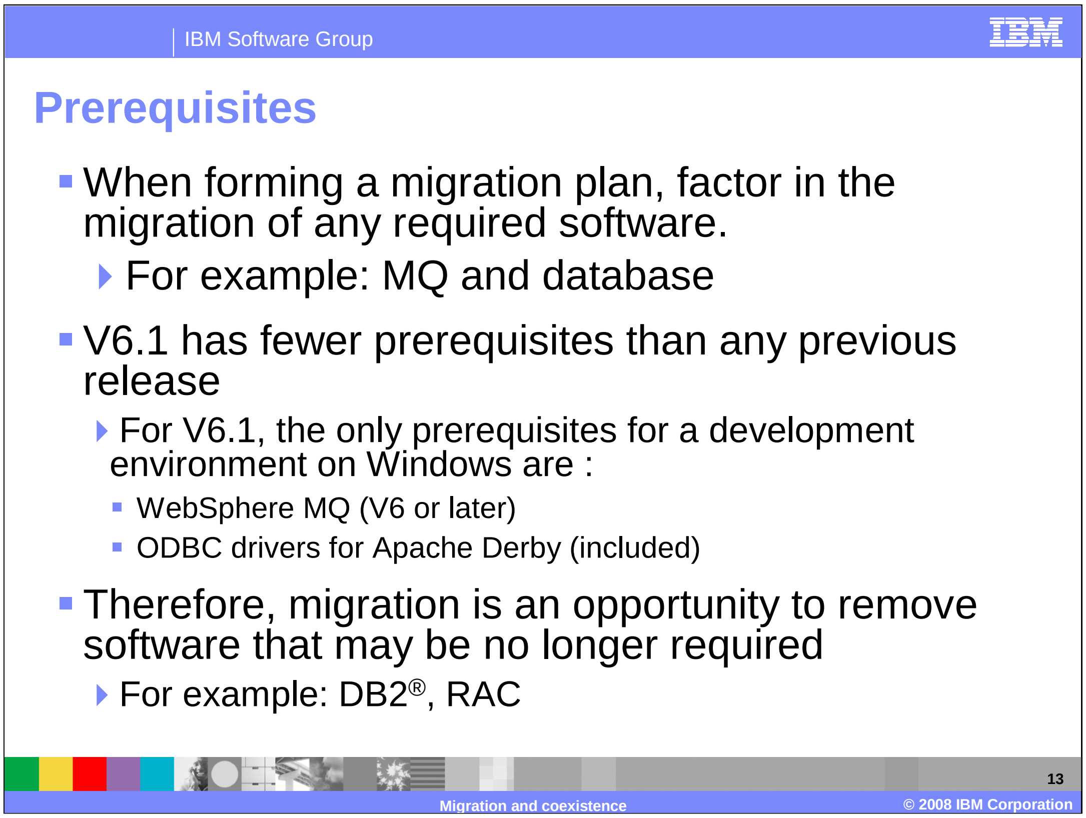
Based on the rules previously described, here is a recommendation for the process of migrating to version 6.1.



What to migrate

- Pre-requisites
- Toolkit and application logic
- Broker
- Configuration manager
- DB instance manager
- User name server

When migrating to version 6.1, there are several components that will need to be migrated. This slide provides a summary, and the next section discusses each of these in turn.



Prerequisites

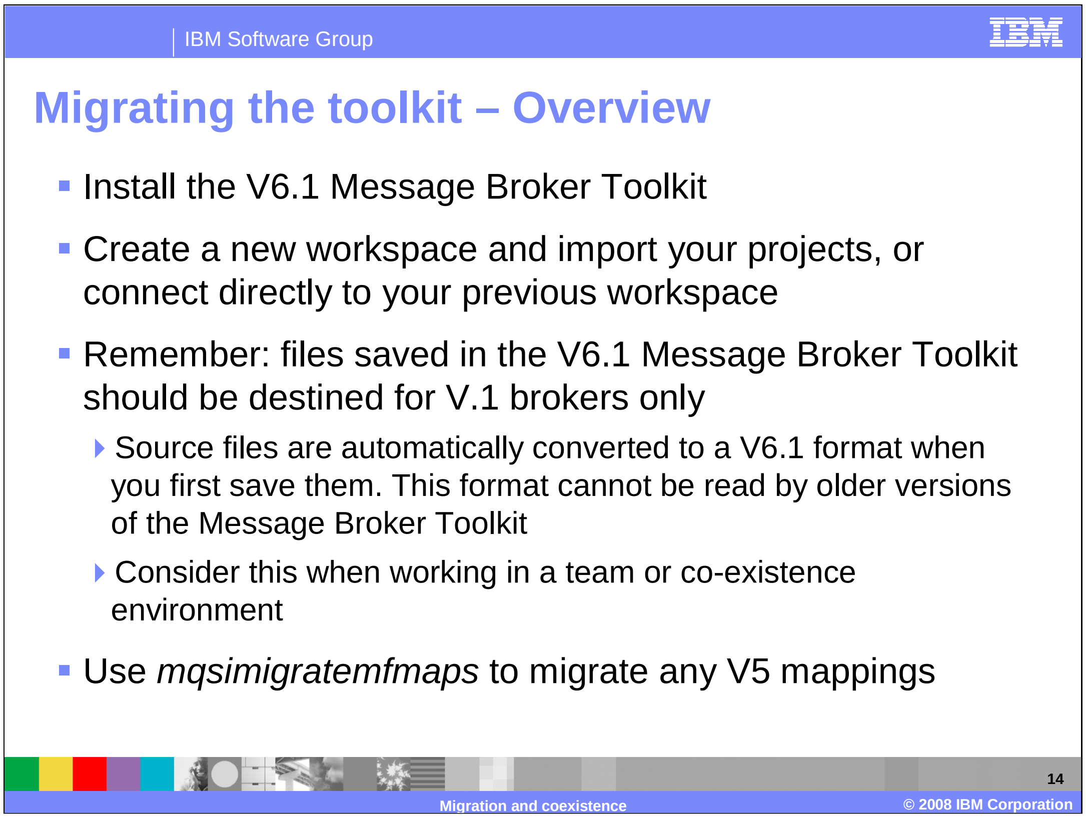
- When forming a migration plan, factor in the migration of any required software.
  - For example: MQ and database
- V6.1 has fewer prerequisites than any previous release
  - For V6.1, the only prerequisites for a development environment on Windows are :
    - WebSphere MQ (V6 or later)
    - ODBC drivers for Apache Derby (included)
- Therefore, migration is an opportunity to remove software that may be no longer required
  - For example: DB2®, RAC

The first thing to consider in the migration process is the set of software that the Message Broker (and your Message Broker applications) requires.

Before you install version 6.1, you must ensure that any required software is at a level that is supported by the new version. When using multiple versions of the Message Broker simultaneously, you should ensure that the versions of pre-requisite software are compatible with both versions of Message Broker that you plan to use. This statement also applies for Operating System levels. For example, Windows 2000 was supported in Message Broker version 5 but it is no longer supported in version 6.1. See the Message Broker Web site for information on supported software levels.

Message Broker version 6.1 has fewer pre-requisite products than ever. This new version has removed a number of these requirements, primarily DB2 for the Configuration Manager and Rational Agent Controller for debugging. You may want to use the migration as an opportunity to remove these additional products.

ODBC Drivers for Apache Derby are included with Message Broker Version 6.1. However, the installer checks for the presence of DB2 on the local machine, and if found will use this instead.



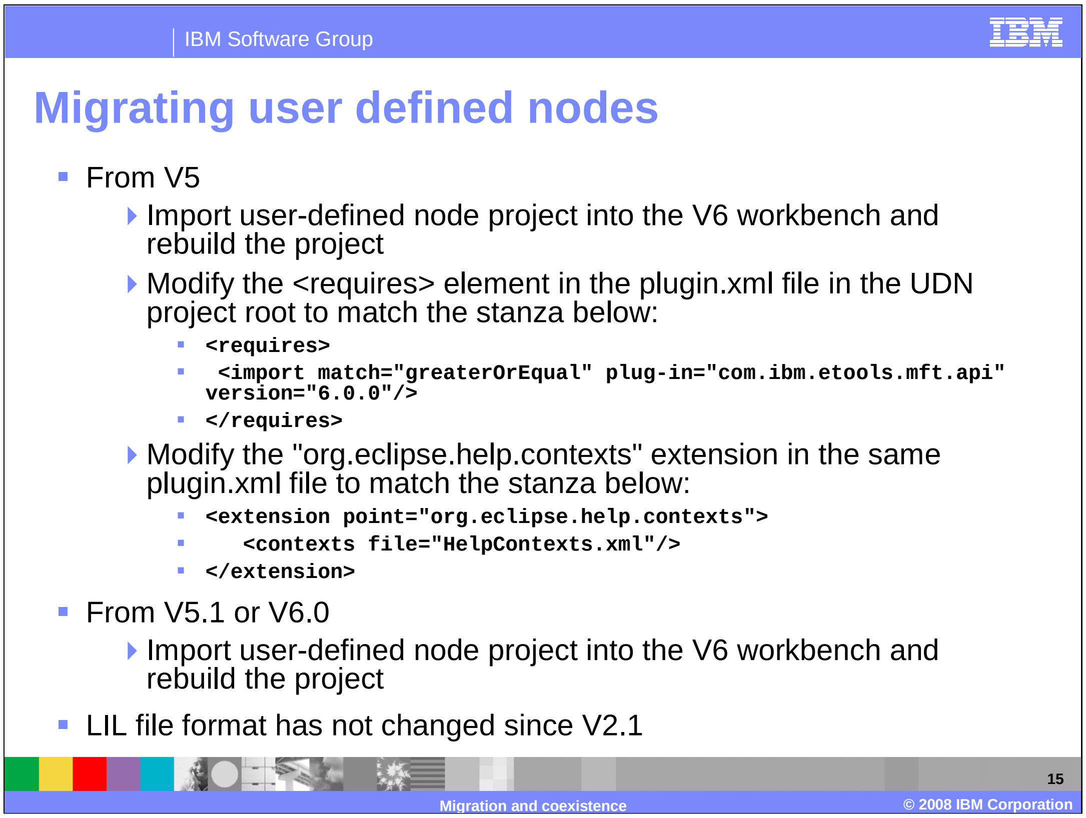
Migrating the toolkit – Overview

- Install the V6.1 Message Broker Toolkit
- Create a new workspace and import your projects, or connect directly to your previous workspace
- Remember: files saved in the V6.1 Message Broker Toolkit should be destined for V.1 brokers only
  - Source files are automatically converted to a V6.1 format when you first save them. This format cannot be read by older versions of the Message Broker Toolkit
  - Consider this when working in a team or co-existence environment
- Use *mqsimigratemfmaps* to migrate any V5 mappings

Migrating your development sets in the Message Broker Toolkit is straightforward. After installing the version 6.1 Toolkit, you must either start a new workspace and import any required artifacts from your old workspace, or you can connect the new Toolkit directly to your existing workspace.

Creating a new workspace is recommended. This is because as soon as you save your existing files in the new workspace you can no longer deploy them to back level brokers, or load them in the previous level of toolkit. You may need to do this while you have back-level brokers to support. The downside of using a second workspace is that you potentially have two copies of your artifacts to maintain – one at version 6.1 and one at the previous version.

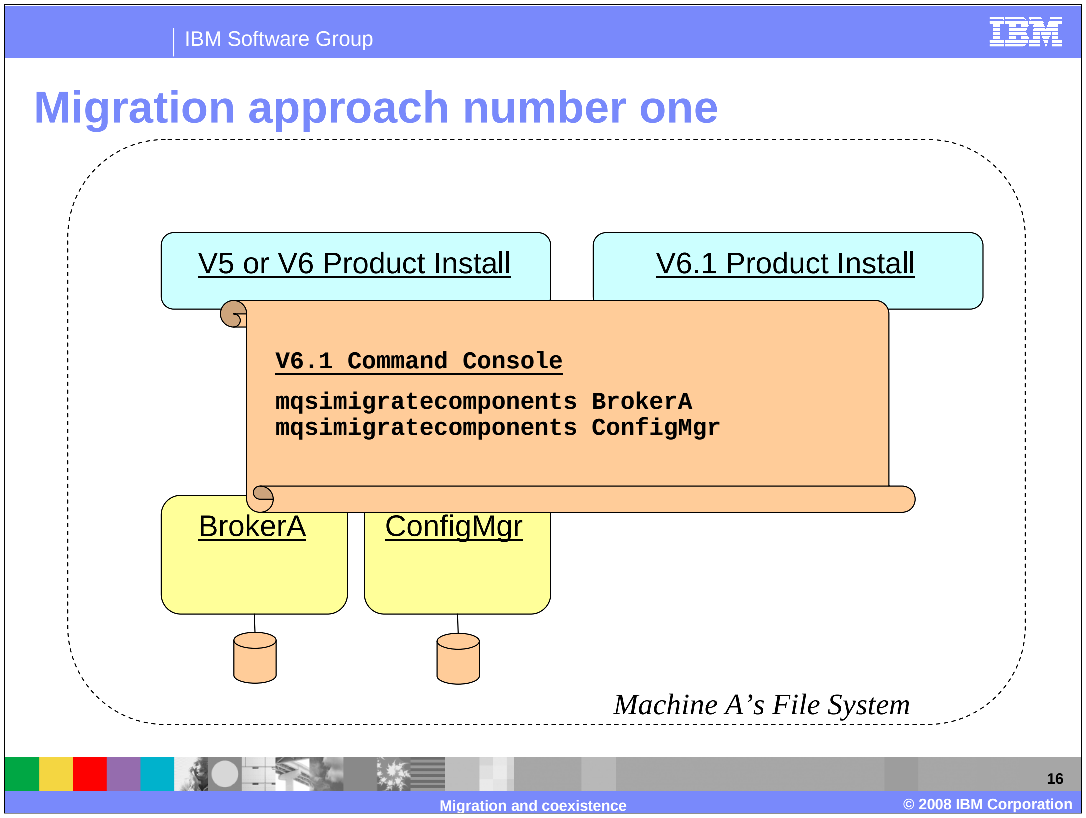
All artifacts are migrated the first time they are saved. The exception is for any “mf-map” files that were generated in version 5, which must be migrated using the “mqsi-migrate-mf-maps” command. Version 6.0 maps are migrated automatically.



Migrating user defined nodes

- From V5
  - Import 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 stanza below:
    - `<requires>`
    - ` <import match="greaterOrEqual" plug-in="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 stanza below:
    - `<extension point="org.eclipse.help.contexts">`
    - `   <contexts file="HelpContexts.xml"/>`
    - `</extension>`
- From V5.1 or V6.0
  - Import user-defined node project into the V6 workbench and rebuild the project
- LIL file format has not changed since V2.1

If you are migrating from version 5 and have user-defined nodes, you need to manually modify the plugin.xml file that is associated with the Message Broker Toolkit User Defined Node project.



IBM Software Group

Migration approach number one

V5 or V6 Product Install

V6.1 Product Install

V6.1 Command Console

```
mqsimigratecomponents BrokerA
mqsimigratecomponents ConfigMgr
```

BrokerA

ConfigMgr

Machine A’s File System

16

Migration and coexistence

The presentation will now discuss migration of the runtime components.

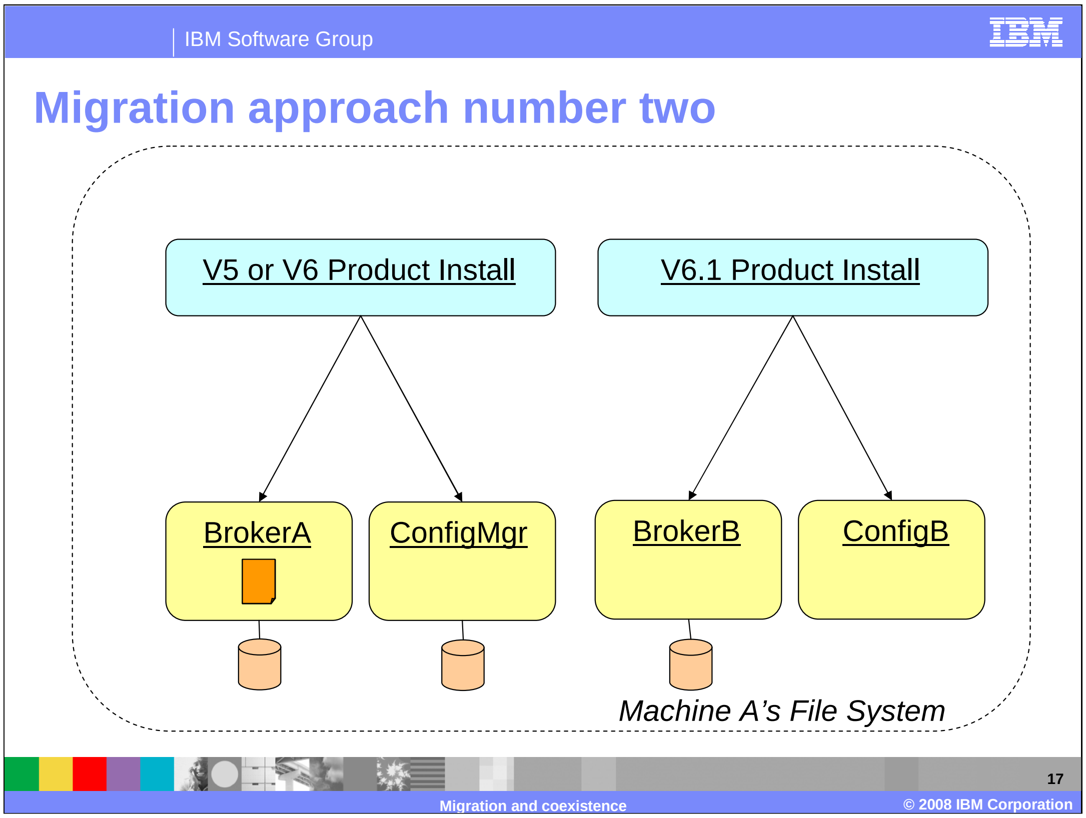
There are two techniques for performing the overall migration of a single machine, and the next two slides describe them both. The scenario will assume that the machine contains a single broker and configuration manager, although the technique can be applied to machines with many brokers and configuration managers, or to machines without one of these components.

Migration of the Message Broker Toolkit will be covered later in the presentation.

The first technique uses the features described on the previous slide. Install the version 6.1 product alongside the previous version of the product, and then open a command console, or provide an appropriate environment to issue broker commands.

In this environment, stop each component and issue the “mqsi-migrate-components” command to migrate it to the new version. At this point the component is now associated with the version 6.1 installation. Starting the component completes the migration.

If you are migrating a configuration manager from version 5, its external database is no longer required.



IBM Software Group

Migration approach number two

V5 or V6 Product Install

V6.1 Product Install

BrokerA

ConfigMgr

BrokerB

ConfigB

Machine A’s File System

17

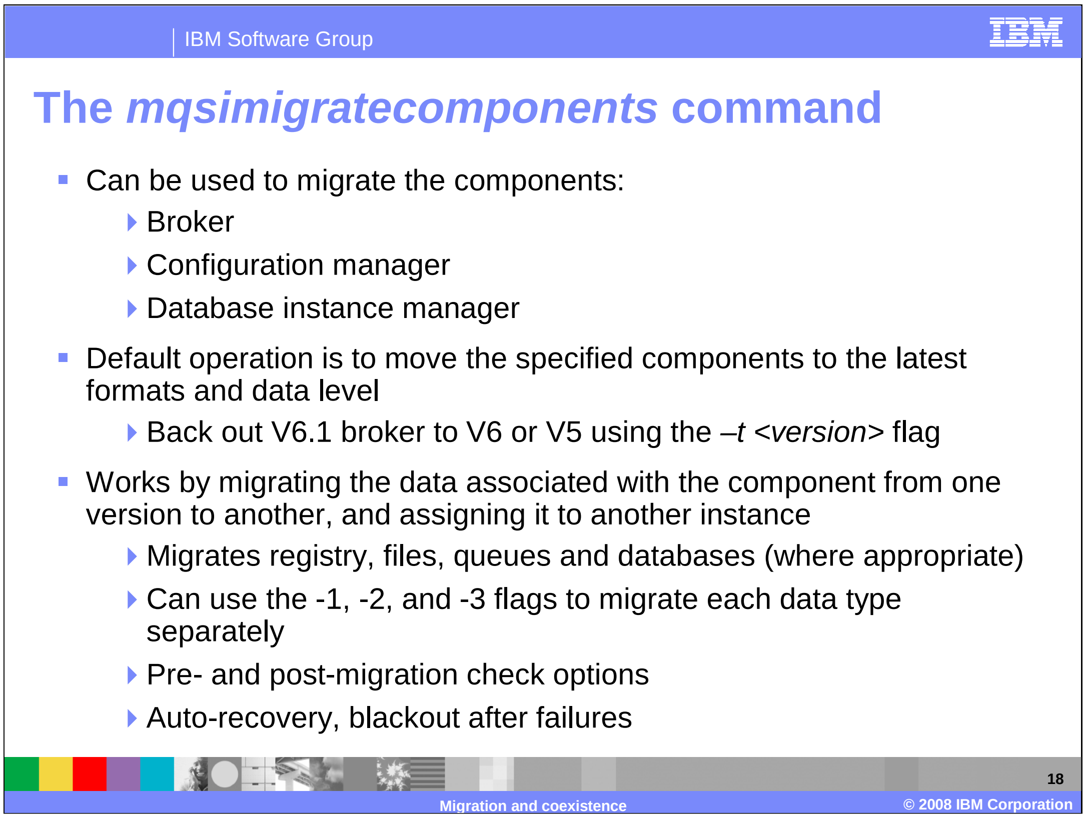
Migration and coexistence

© 2008 IBM Corporation

This slide shows the second approach for migration of a single machine.

You can install version 6.1 alongside the previous version as before, but then create completely new components that are associated with that new installation. Then customize the broker archive files that are associated with the old broker for use in the new version 6.1 environment and deploy them to the new broker. Once you are happy that the message flows are working correctly, the old components can be stopped and removed.

The second technique uses different component names, and hence may require reconfiguration of client applications. However, the advantage of this approach is that the existing brokers do not need to be stopped in order for migration to occur.



IBM Software Group

The *mqsimigratecomponents* command

- Can be used to migrate the components:
  - Broker
  - Configuration manager
  - Database instance manager
- Default operation is to move the specified components to the latest formats and data level
  - Back out V6.1 broker to V6 or V5 using the *–t <version>* flag
- Works by migrating the data associated with the component from one version to another, and assigning it to another instance
  - Migrates registry, files, queues and databases (where appropriate)
  - Can use the -1, -2, and -3 flags to migrate each data type separately
  - Pre- and post-migration check options
  - Auto-recovery, blackout after failures

18

Migration and coexistence

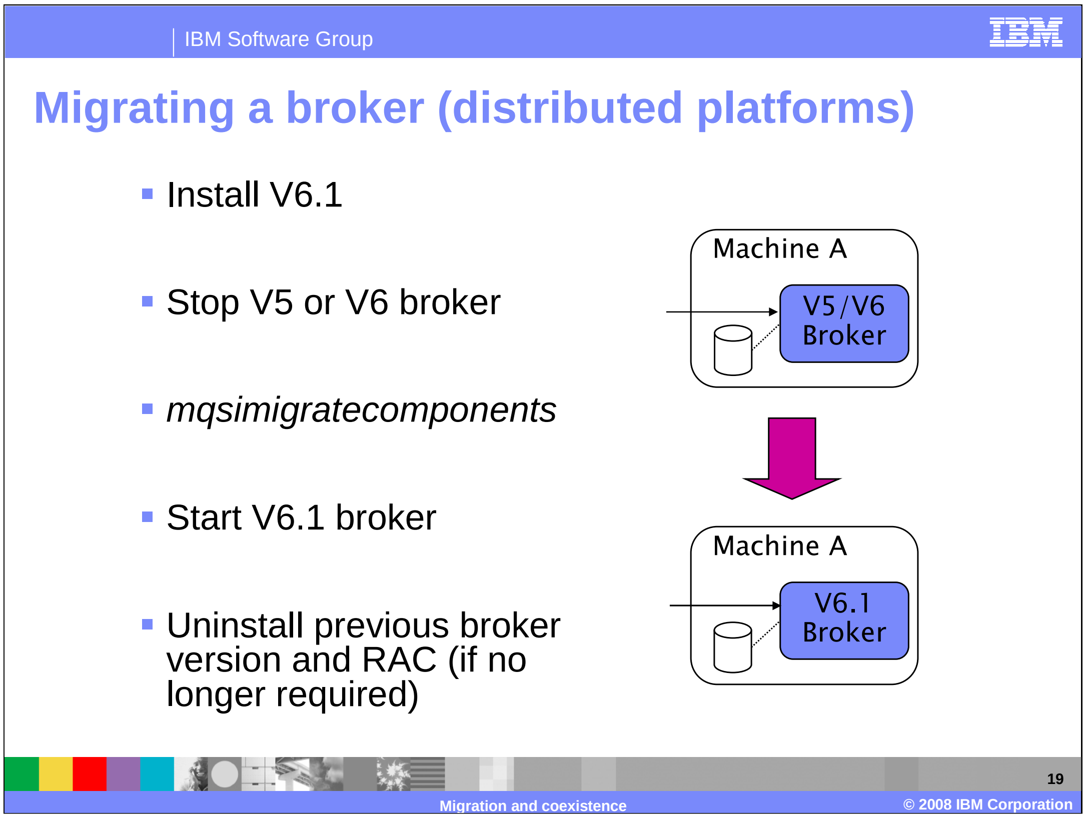
© 2008 IBM Corporation

This slide discusses the “*mqsi-migrate-components*” command, which is used to move brokers, configuration managers and database instance managers between versions. By default, it will detect the current version of the component it is trying to migrate and move it up to the most recent version, in this case version 6.1.

It can also be used against components that are already at the 6.1 level, if the “–t” flag is specified. This has the effect of reverting that component to the version specified. For example, specifying “–t 5.0.0.4” will revert a broker to 5.0 fix pack 4. The command will only modify the data associated with the component so that it is compatible with the version specified. No product code is changed, and so a reversion to version 5.0 fix pack 4 assumes that this level is actually installed on the local system. It is not possible to revert a configuration manager to version 5.

Four artifacts are potentially associated with each runtime component, and each of these is migrated using this command. The four artifacts are: registry, files, queues and databases.

Some business environments place restrictions on which users can modify which artifacts. For example, the database administrator may not be able to modify MQ objects. Because of this, the “-1”, “-2”, and “-3” flags can be used to migrate only parts of the component at any one time. “-1” migrates the file system and registry. “-2” migrates MQ artifacts. “-3” migrates database artifacts. Only when all of these parts have been migrated will the component be usable at the target level. If none of these flags are specified, everything is migrated.



IBM Software Group

Migrating a broker (distributed platforms)

- Install V6.1
- Stop V5 or V6 broker
- *mqsimigratecomponents*
- Start V6.1 broker
- Uninstall previous broker version and RAC (if no longer required)

Machine A

V5/V6 Broker

Machine A

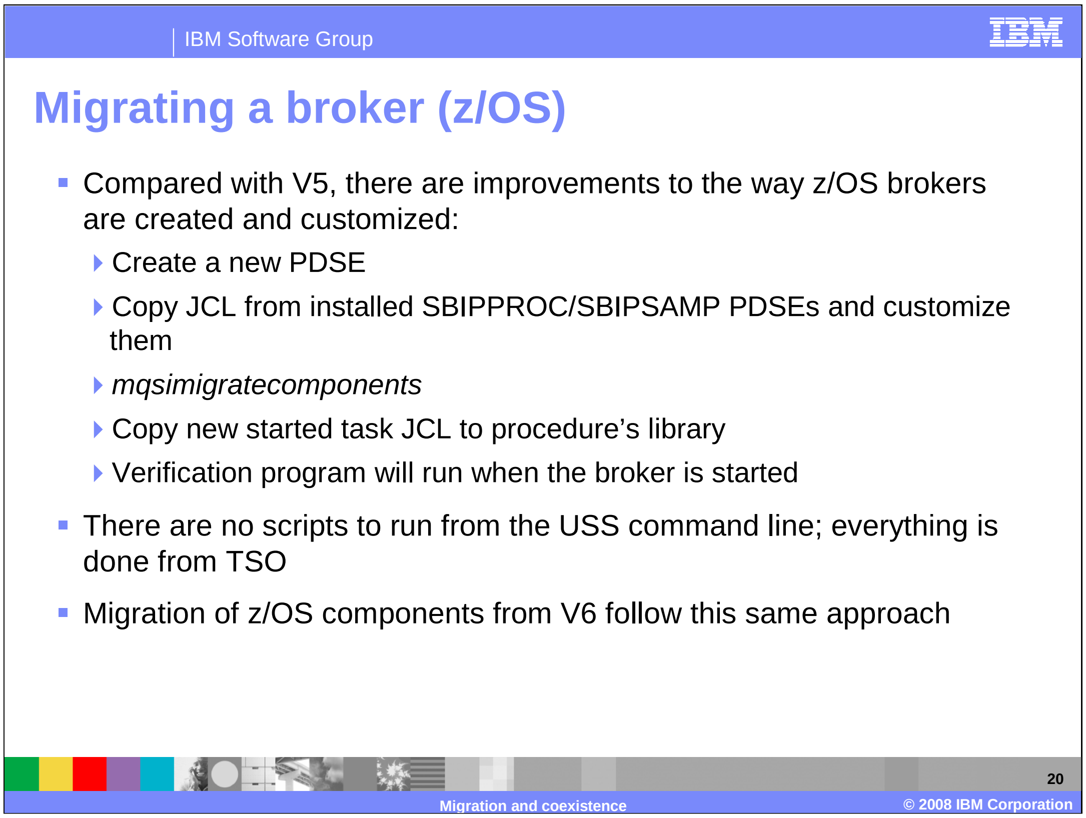
V6.1 Broker

19

Migration and coexistence

© 2008 IBM Corporation

This slide describes the general approach for migrating brokers on distributed platforms.



IBM Software Group

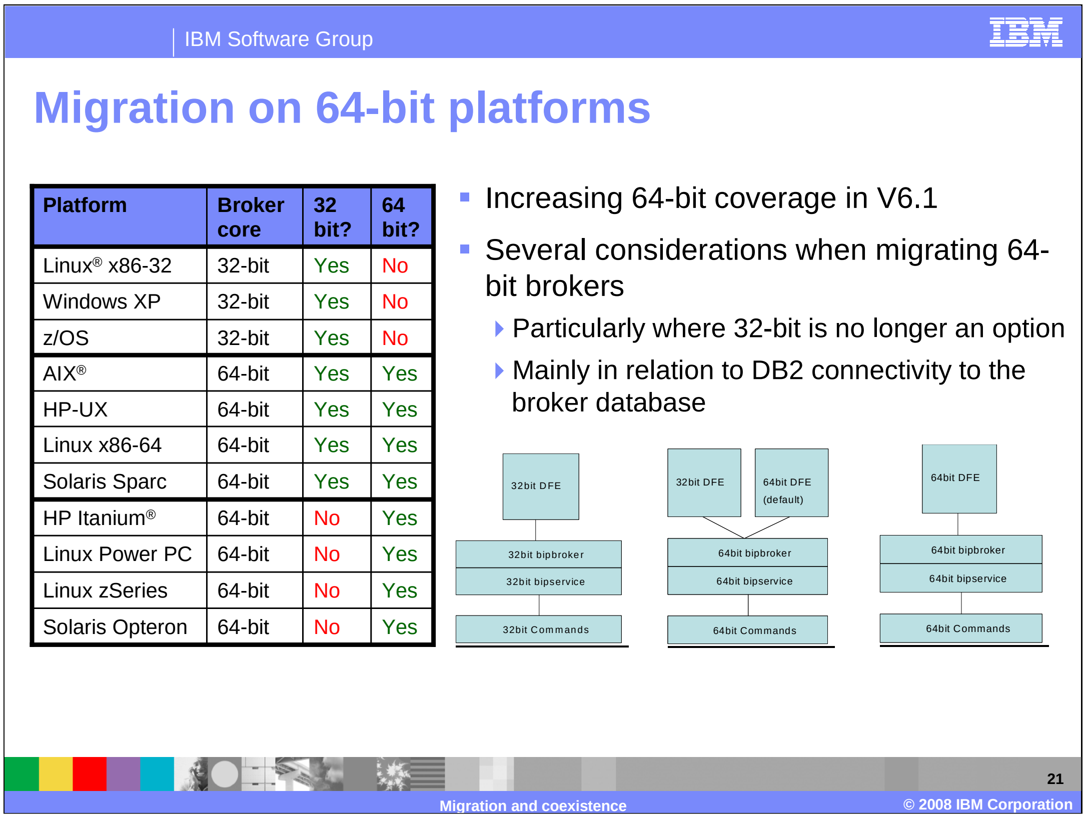
Migrating a broker (z/OS)

- Compared with 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
  - *mqsimigratecomponents*
  - 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
- Migration of z/OS components from V6 follow this same approach

20

Migration and coexistence

Migrating brokers on z/OS follows uses the same general technique as distributed platforms, with the few additional steps described here.



IBM Software Group

Migration on 64-bit platforms

Platform	Broker core	32 bit?	64 bit?
Linux® x86-32	32-bit	Yes	No
Windows XP	32-bit	Yes	No
z/OS	32-bit	Yes	No
AIX®	64-bit	Yes	Yes
HP-UX	64-bit	Yes	Yes
Linux x86-64	64-bit	Yes	Yes
Solaris Sparc	64-bit	Yes	Yes
HP Itanium®	64-bit	No	Yes
Linux Power PC	64-bit	No	Yes
Linux zSeries	64-bit	No	Yes
Solaris Opteron	64-bit	No	Yes

- Increasing 64-bit coverage in V6.1
- Several considerations when migrating 64-bit brokers
  - Particularly where 32-bit is no longer an option
  - Mainly in relation to DB2 connectivity to the broker database

32bit DFE
32bit bipbroker
32bit bipservice
32bit Commands

32bit DFE
64bit DFE (default)
64bit bipbroker
64bit bipservice
64bit Commands

64bit DFE
64bit bipbroker
64bit bipservice
64bit Commands

21

Migration and coexistence

There are three types of platform for Message Broker version 6.1.

First, platforms that are completely 32-bit. These have 32-bit commands, a 32-bit broker core (bip-broker and bip-service processes) and a 32-bit Data-Flow-Engine, representing the execution groups. *(Note to presenter: "bip" is pronounced "bip").*

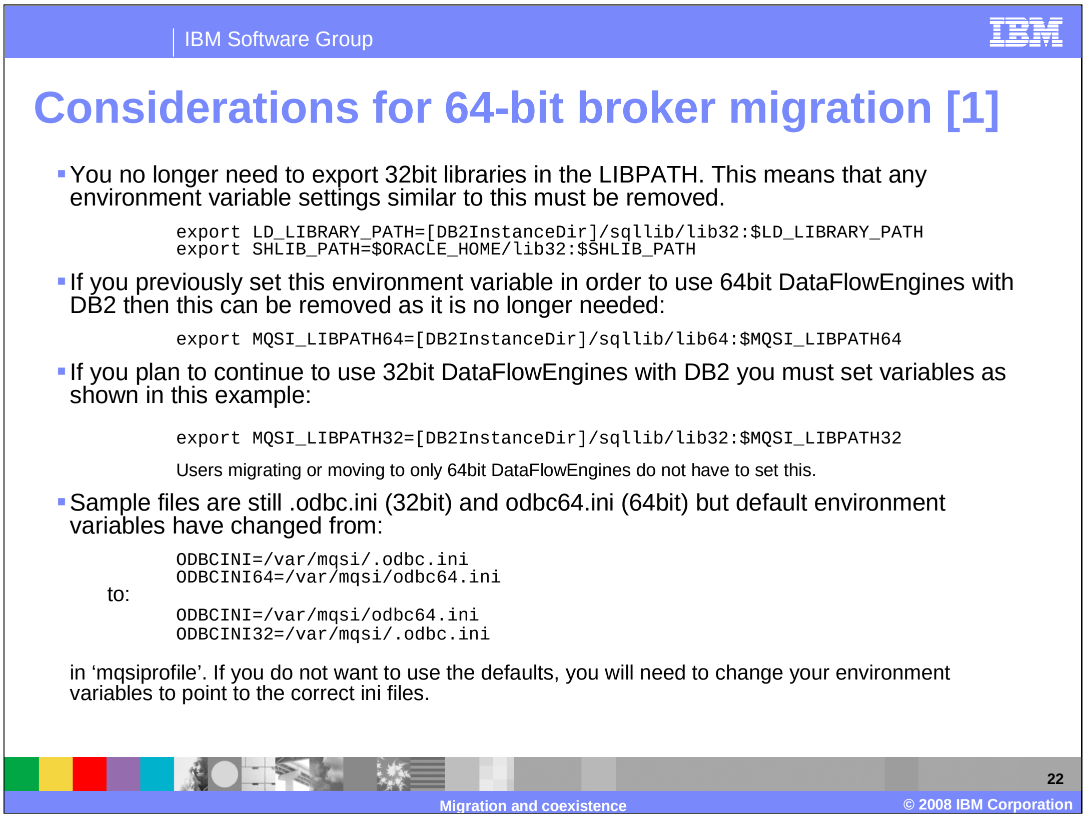
Second, platforms that have 32-bit and 64-bit support. These have 64-bit commands, a 64-bit broker core and both 32 and 64-bit Data-Flow-Engines.

Third, platforms that are 64-bit only. These have 64-bit commands, a 64-bit broker core and only 64-bit Data-Flow-Engines.

There are several additional steps when migrating on 64-bit platforms, particularly when migrating from a 32-bit system. The next two slides describe these steps.

Use the command "db2level" from a DB2 command prompt to indicate the mode of DB2 being used.

There are no known problems with other database managers.



IBM Software Group

Considerations for 64-bit broker migration [1]

- You no longer need to export 32bit libraries in the LIBPATH. This means that any environment variable settings similar to this must be removed.

```
export LD_LIBRARY_PATH=[DB2InstanceDir]/sqllib/lib32:$LD_LIBRARY_PATH
export SHLIB_PATH=$ORACLE_HOME/lib32:$SHLIB_PATH
```

- If you previously set this environment variable in order to use 64bit DataFlowEngines with DB2 then this can be removed as it is no longer needed:

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

- If you plan to continue to use 32bit DataFlowEngines with DB2 you must set variables as shown in this example:

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

Users migrating or moving to only 64bit DataFlowEngines do not have to set this.

- Sample files are still .odbc.ini (32bit) and odbc64.ini (64bit) but default environment variables have changed from:

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

to:

```
ODBCINI=/var/mqsi/odbc64.ini
ODBCINI32=/var/mqsi/.odbc.ini
```

in ‘mqsiprofile’. If you do not want to use the defaults, you will need to change your environment variables to point to the correct ini files.

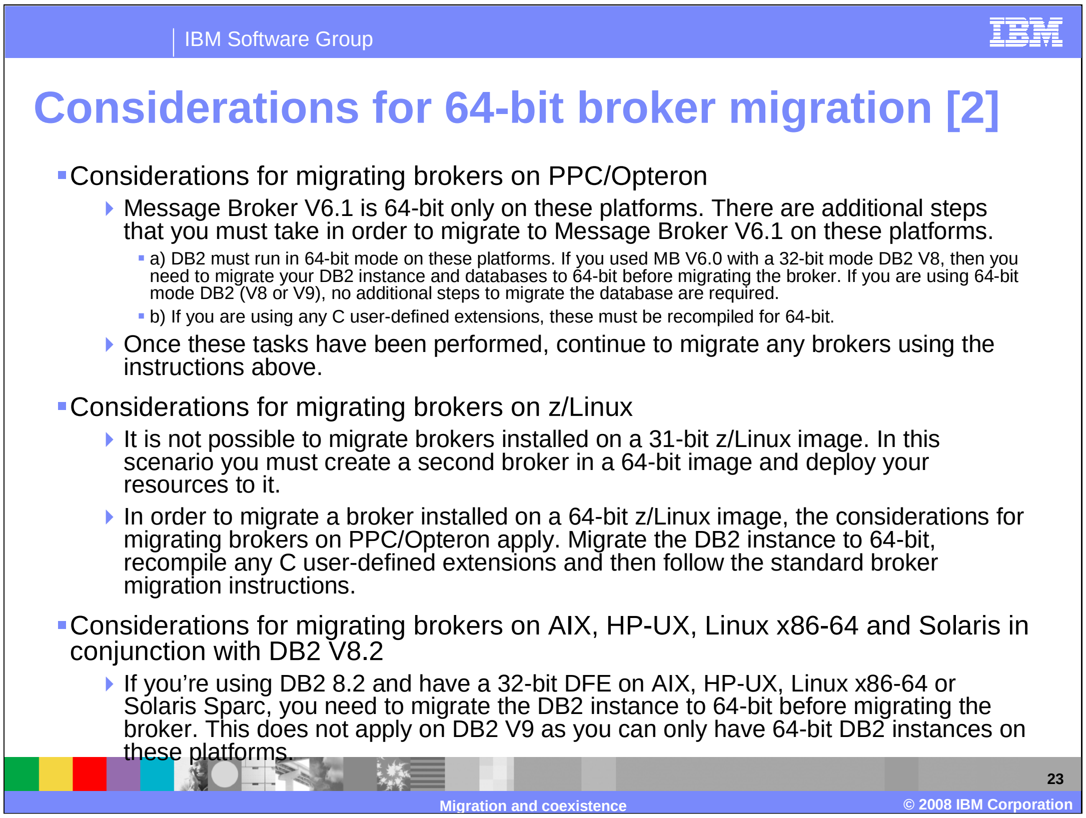
22

Migration and coexistence

This slide describes some of the additional steps required when migrating on 64-bit platforms.

You should take the time to study this slide. In particular, the ODBC settings are important, since the default setting is now for a 64-bit system on UNIX systems. If you are moving from a 32-bit to a 64-bit environment, this will have implications on your “ODBC-INI” setting.

For example, on AIX, the default has changed from 32-bit to 64-bit processing.



IBM Software Group

Considerations for 64-bit broker migration [2]

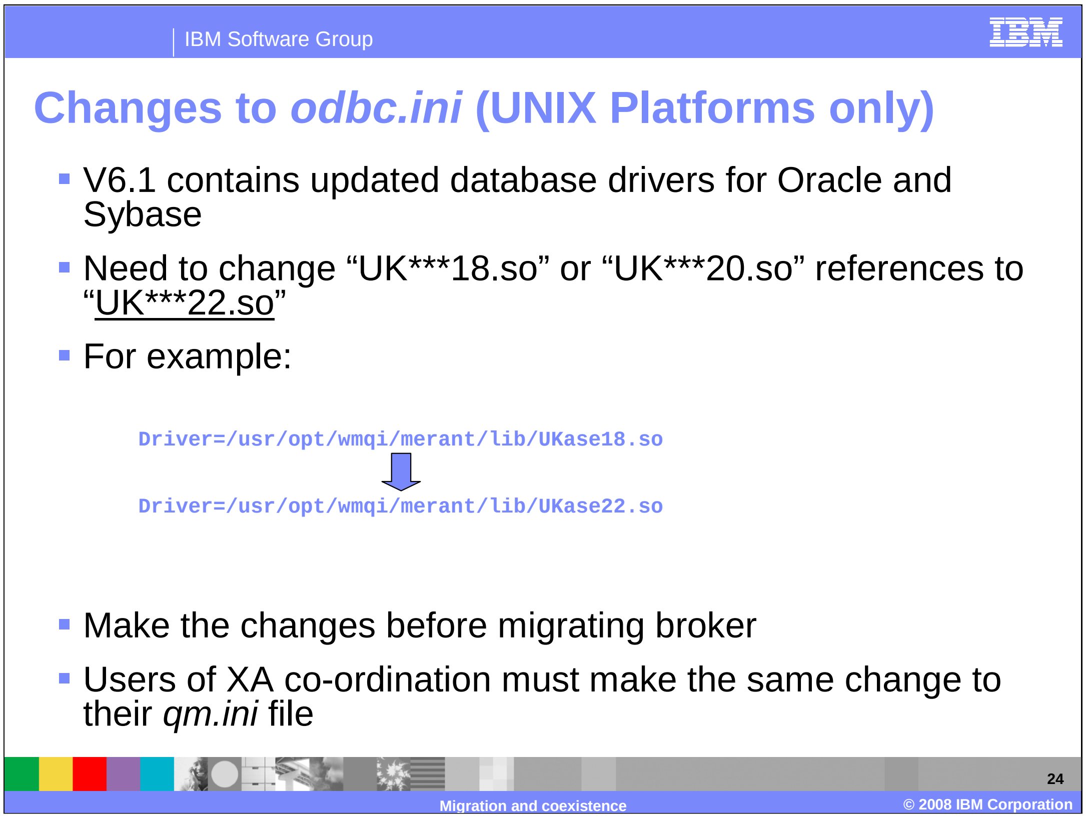
- Considerations for migrating brokers on PPC/Opteron
  - Message Broker V6.1 is 64-bit only on these platforms. There are additional steps that you must take in order to migrate to Message Broker V6.1 on these platforms.
    - a) DB2 must run in 64-bit mode on these platforms. If you used MB V6.0 with a 32-bit mode DB2 V8, then you need to migrate your DB2 instance and databases to 64-bit before migrating the broker. If you are using 64-bit mode DB2 (V8 or V9), no additional steps to migrate the database are required.
    - b) If you are using any C user-defined extensions, these must be recompiled for 64-bit.
  - Once these tasks have been performed, continue to migrate any brokers using the instructions above.
- Considerations for migrating brokers on z/Linux
  - It is not possible to migrate brokers installed on a 31-bit z/Linux image. In this scenario you must create a second broker in a 64-bit image and deploy your resources to it.
  - In order to migrate a broker installed on a 64-bit z/Linux image, the considerations for migrating brokers on PPC/Opteron apply. Migrate the DB2 instance to 64-bit, recompile any C user-defined extensions and then follow the standard broker migration instructions.
- Considerations for migrating brokers on AIX, HP-UX, Linux x86-64 and Solaris in conjunction with DB2 V8.2
  - If you're using DB2 8.2 and have a 32-bit DFE on AIX, HP-UX, Linux x86-64 or Solaris Sparc, you need to migrate the DB2 instance to 64-bit before migrating the broker. This does not apply on DB2 V9 as you can only have 64-bit DB2 instances on these platforms.

23

Migration and coexistence

© 2008 IBM Corporation

This slide describes the other additional steps required when migrating on 64-bit platforms.



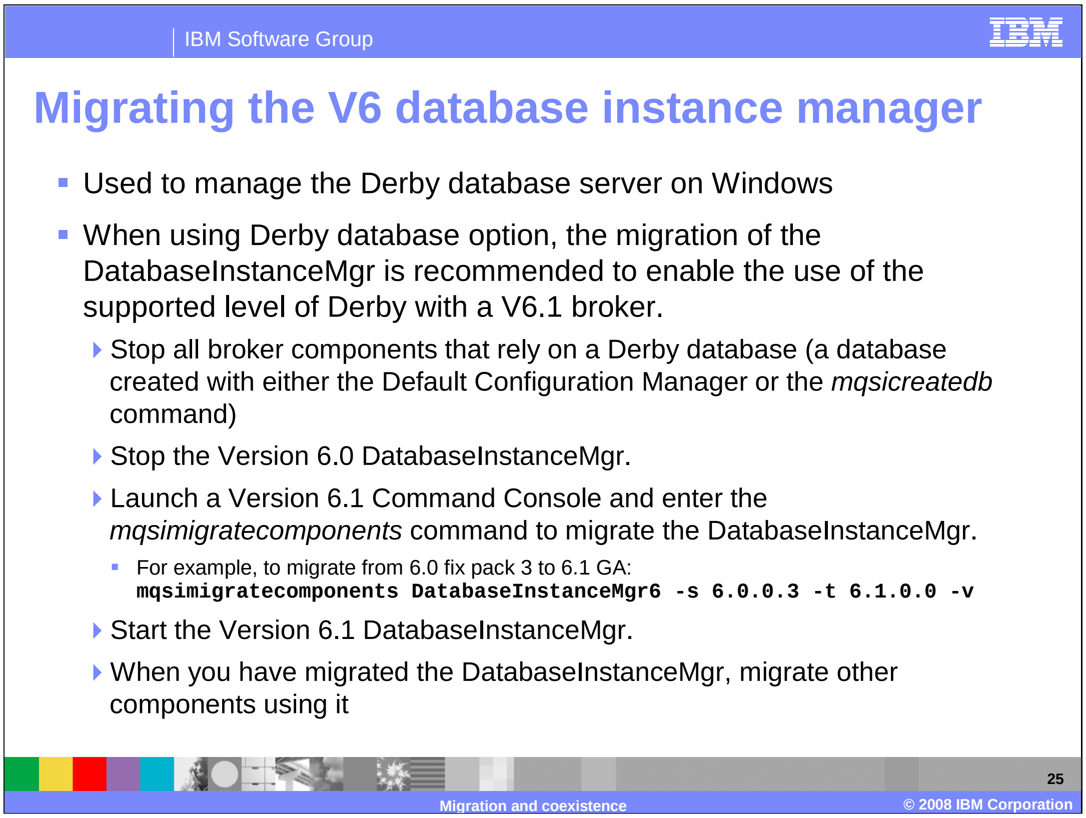
Changes to *odbc.ini* (UNIX Platforms only)

- V6.1 contains updated database drivers for Oracle and Sybase
- Need to change “UK***18.so” or “UK***20.so” references to “UK***22.so”
- For example:

```
Driver=/usr/opt/wmqi/merant/lib/UKase18.so
                    ⇩
Driver=/usr/opt/wmqi/merant/lib/UKase22.so
```

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

This slide describes the changes required to the odbc.ini file when connecting to databases from a version 6.1 broker.



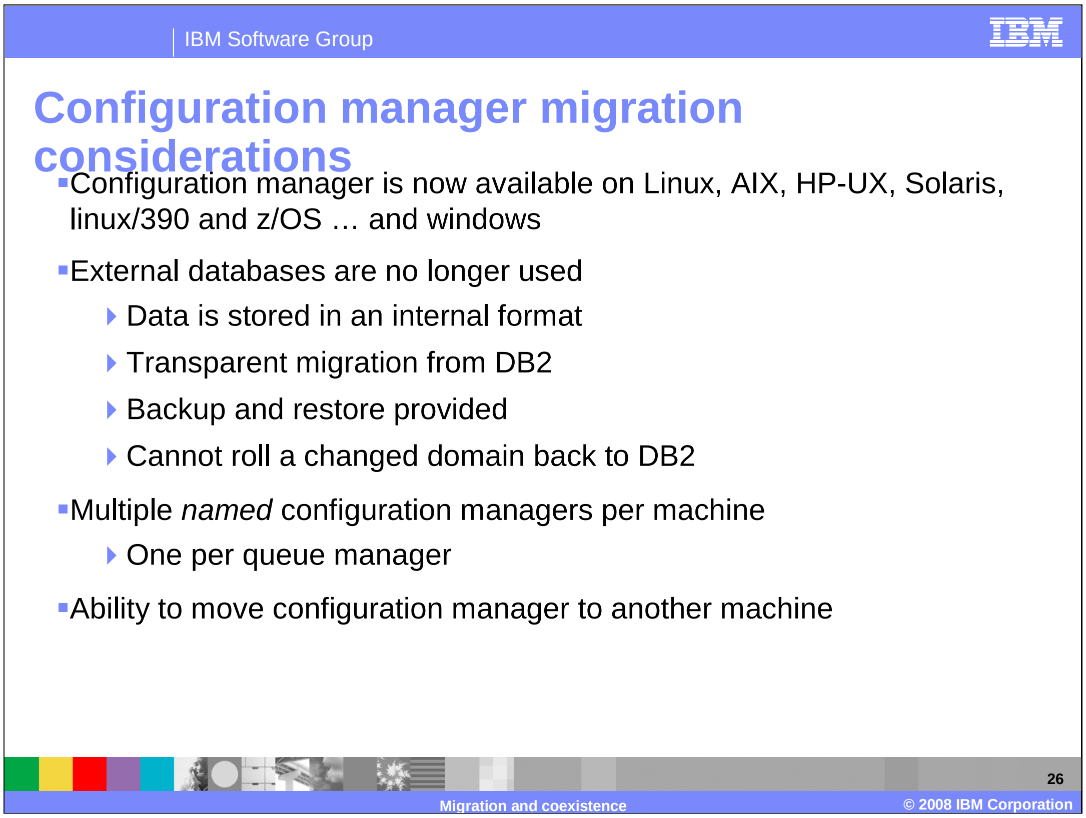
Migrating the V6 database instance manager

- Used to manage the Derby database server on Windows
- When using Derby database option, the migration of the DatabaseInstanceMgr is recommended to enable the use of the supported level of Derby with a V6.1 broker.
  - Stop all broker components that rely on a Derby database (a database created with either the Default Configuration Manager or the *mqsicreatedb* command)
  - Stop the Version 6.0 DatabaseInstanceMgr.
  - Launch a Version 6.1 Command Console and enter the *mqsimigratecomponents* command to migrate the DatabaseInstanceMgr.
    - For example, to migrate from 6.0 fix pack 3 to 6.1 GA:

```
mqsimigratecomponents DatabaseInstanceMgr6 -s 6.0.0.3 -t 6.1.0.0 -v
```

  - Start the Version 6.1 DatabaseInstanceMgr.
  - When you have migrated the DatabaseInstanceMgr, migrate other components using it

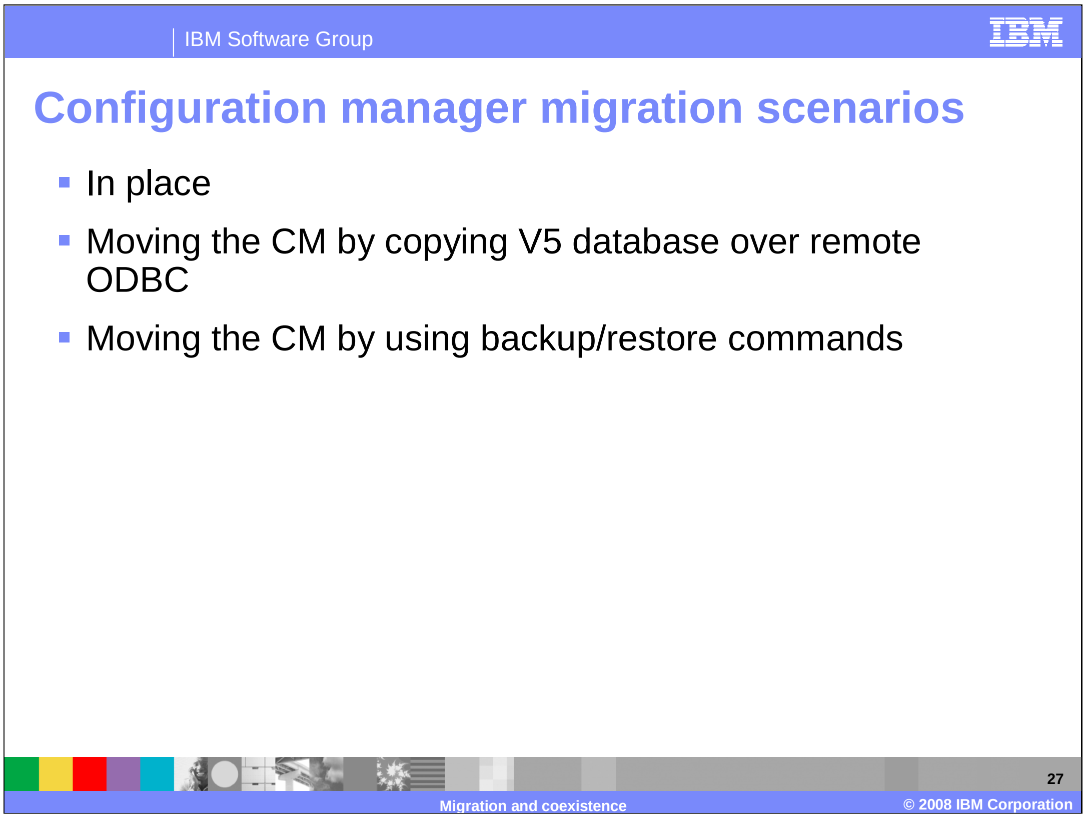
If you are planning to migrate version 6.0 brokers that use a Derby database, then it is recommended that you migrate the Database Instance Manager. This will ensure that you use the updated level of Derby that comes with Message Broker version 6.1.



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

The presentation now discusses migration of the configuration manager. Significant improvements were made to the configuration manager in version 6.0, so if you’re migrating from version 5, there are several things to bear in mind.



Configuration manager migration scenarios

- In place
- Moving the CM by copying V5 database over remote ODBC
- Moving the CM by using backup/restore commands

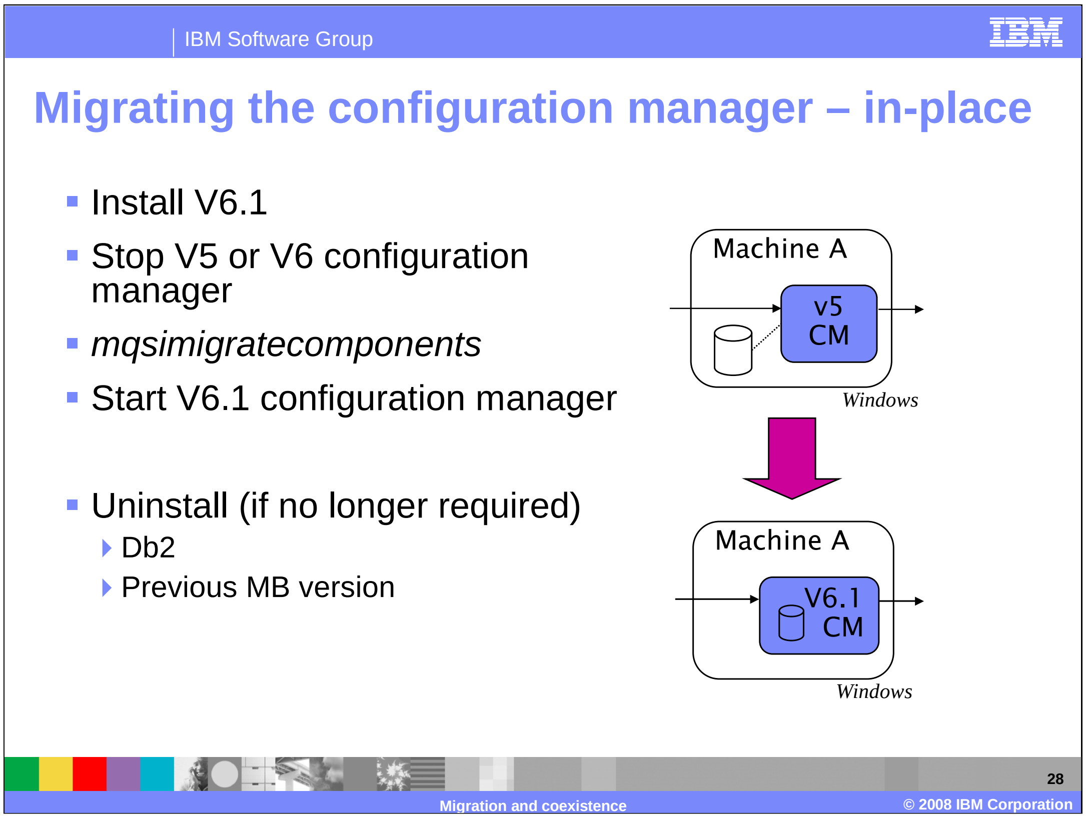
There are three ways of migrating the configuration manager.

The first is the “in place” migration. In this scenario, you have a configuration manager at version 5 or 6.0, and you are going to migrate it directly to version 6.1.

The second scenario is if you are moving from version 5 and you are going to move the Windows configuration manager onto a different machine. This may be on to a UNIX, Linux or z/OS platform, where you have DB2 client libraries available on the target platform.

The third scenario makes use of the “mqsibackupconfigmgr” and “mqsirestoreconfigmgr” commands, and can also be used to move a configuration manager onto a different machine.

The next few slides discuss these scenarios in more detail.



Migrating the configuration manager – in-place

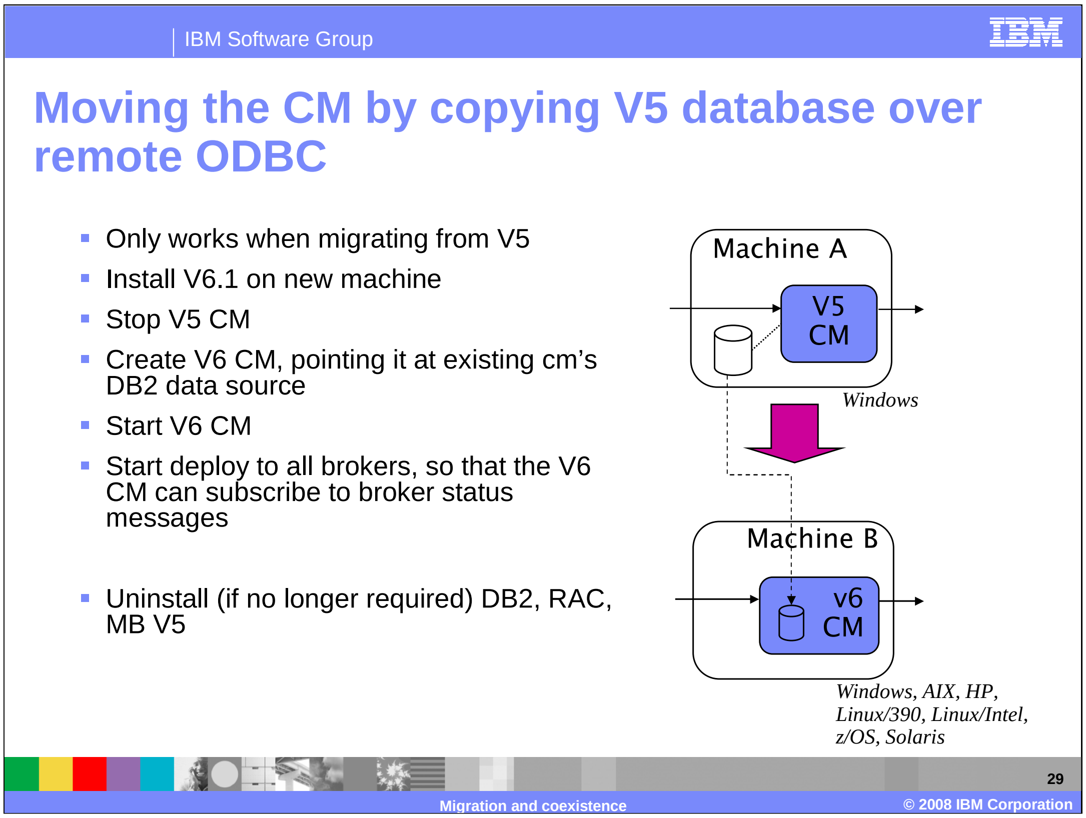
- Install V6.1
- Stop V5 or V6 configuration manager
- *mqsimigratecomponents*
- Start V6.1 configuration manager
- Uninstall (if no longer required)
  - Db2
  - Previous MB version

Machine A
v5
CM
Windows
Machine A
V6.1
CM
Windows

This is easiest way of migrating the configuration manager. To do this, you should perform these tasks: First, install version 6.1 on the configuration manager machine.

Second, stop the old configuration manager. Third, start a version 6.1 environment profile, and issue the “mqsimigratecomponents” command and then restart the configuration manager

If you are migrating from version 5, note that the migration from the DB2 database occurs when the version 6.1 configuration manager starts for the first time. This means that you must not uninstall DB2 until you have at least verified that the version 6.1 configuration manager has transferred this information and is successfully working. Normally, this takes only a minute or so, but you can see details of the database migration progress by reviewing configuration manager messages in the system log.



Moving the CM by copying V5 database over remote ODBC

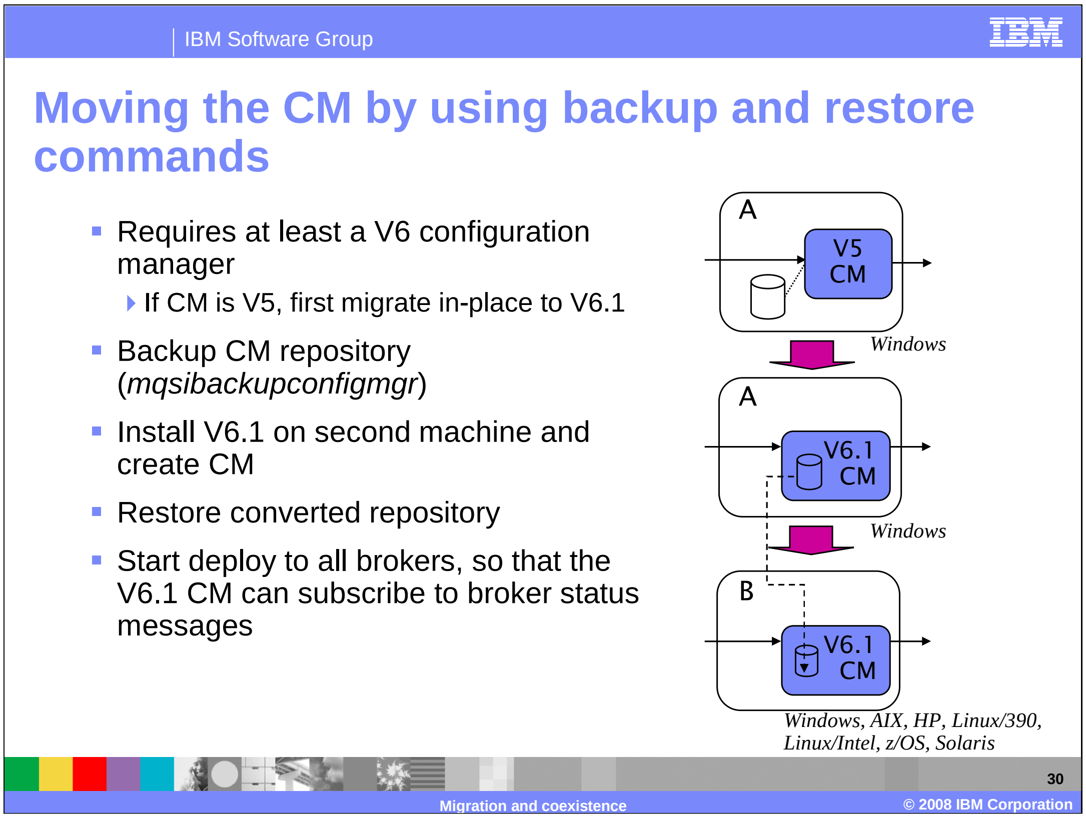
- Only works when migrating from V5
- Install V6.1 on new machine
- Stop V5 CM
- Create V6 CM, pointing it at existing cm’s DB2 data source
- Start V6 CM
- Start deploy to all brokers, so that the V6 CM can subscribe to broker status messages
- Uninstall (if no longer required) DB2, RAC, MB V5

Machine A
V5 CM
Windows
Machine B
v6 CM
Windows, AIX, HP, Linux/390, Linux/Intel, z/OS, Solaris

The version 6.1 “mqsi-create-config-mgr” command includes “–n”, “ –u” and “–p” flags, which are the optional connection details to a DB2 database. If specified, the new configuration manager will be primed with the configuration manager domain information at this location. Using these flags, it is possible to move the version 5 configuration manager on to a second machine by having it extract the information over a remote ODBC connection.

To do this, follow the steps described in this slide. The deployment step is required in order to tell all the brokers in the domain that it must now report status information to a new configuration manager.

As the version 6 configuration manager does not use DB2, this technique cannot be used to migrate a version 6 configuration manager to version 6.1.



IBM Software Group

Moving the CM by using backup and restore commands

- Requires at least a V6 configuration manager
  - If CM is V5, first migrate in-place to V6.1
- Backup CM repository (*mqsibackupconfigmgr*)
- Install V6.1 on second machine and create CM
- Restore converted repository
- Start deploy to all brokers, so that the V6.1 CM can subscribe to broker status messages

A
V5 CM
Windows
A
V6.1 CM
Windows
B
V6.1 CM
Windows, AIX, HP, Linux/390, Linux/Intel, z/OS, Solaris

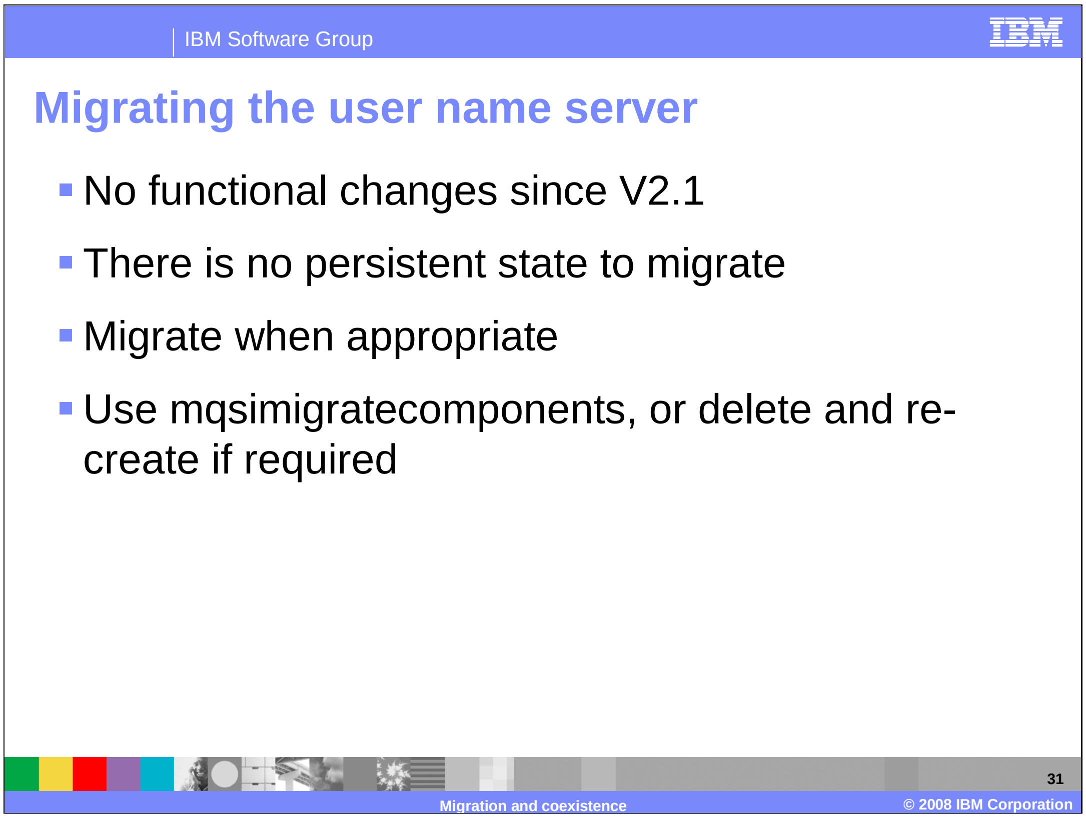
30

Migration and coexistence

Both version 6.0 and 6.1 of Message Broker include commands for backing up and restoring the configuration manager. By backing up a version 6 configuration manager and restoring it into a newly created version 6.1 configuration manager, you can easily migrate a configuration manager. This can be on the same or a different computer.

You can also use this technique to move a version 6.1 configuration manager onto a different machine. However, if you are going to migrate from a version 5 configuration manager, you must first migrate your configuration manager in place to version 6.0 or 6.1. This will allow you to make use of the “mqsi-backup-config-mgr” and “mqsi-restore-config-mgr” commands.

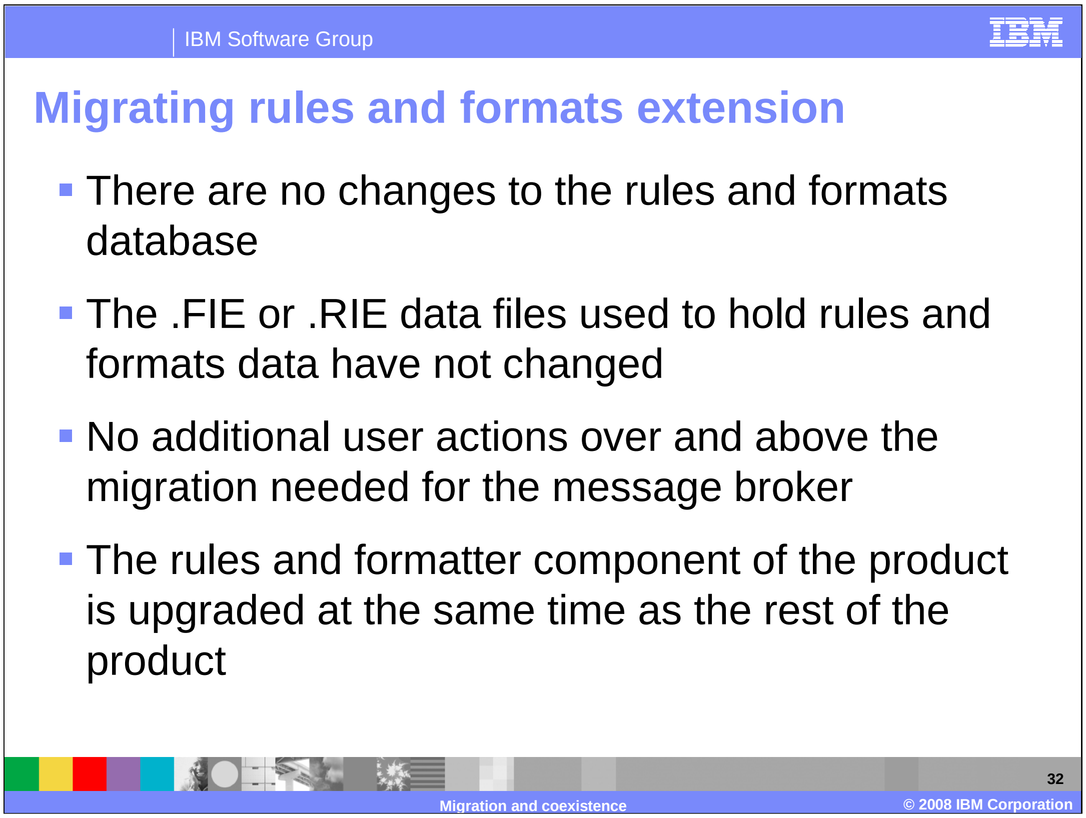
As before, if you are moving the configuration manager on to a different machine or queue manager you need to initiate a deployment to all brokers. This will enable them to start publishing status information to a new configuration manager.



Migrating the user name server

- No functional changes since V2.1
- There is no persistent state to migrate
- Migrate when appropriate
- Use mqsimigratecomponents, or delete and re-create if required

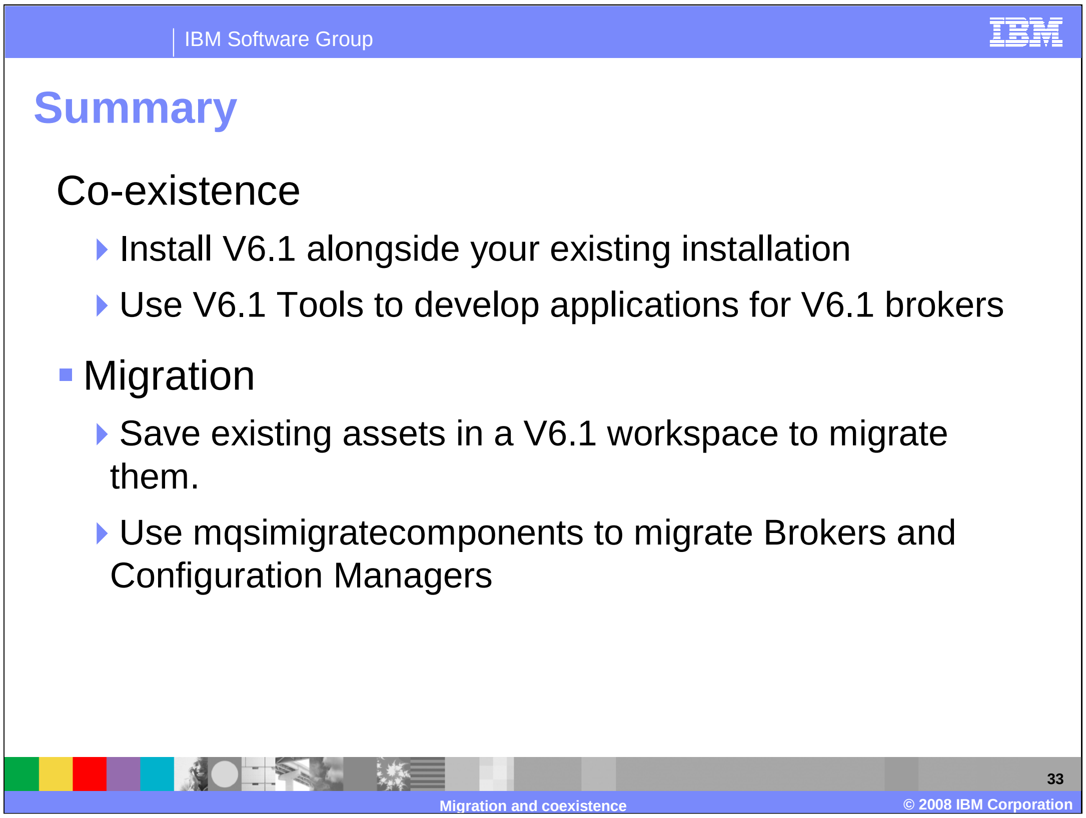
If you use a user name server, migrate it to version 6.1 when convenient.



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

Since version 5 there have been no significant changes to the rules and formatter extension.



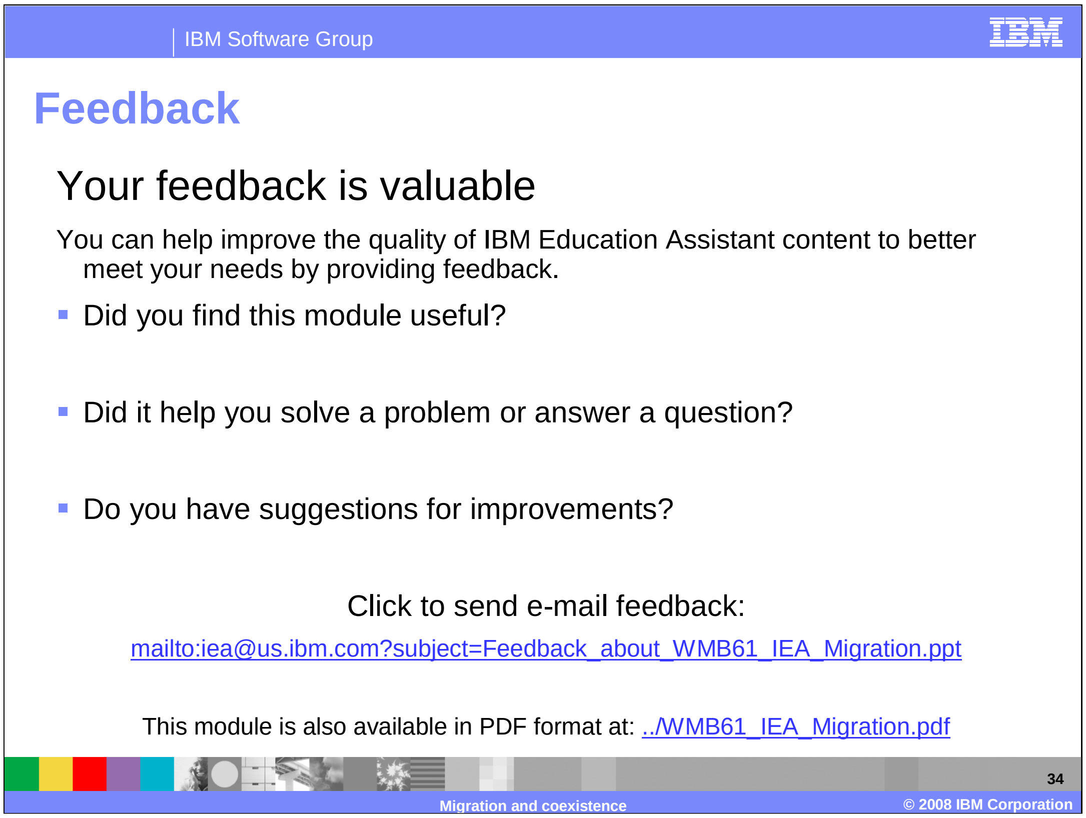
Summary

Co-existence

- Install V6.1 alongside your existing installation
- Use V6.1 Tools to develop applications for V6.1 brokers

- Migration
  - Save existing assets in a V6.1 workspace to migrate them.
  - Use mqsimigratecomponents to migrate Brokers and Configuration Managers

This slide summarizes the co-existence capabilities of Message Broker version 6.1 and describes the key techniques for migrating your components and development artifacts.



Feedback

Your feedback is valuable

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

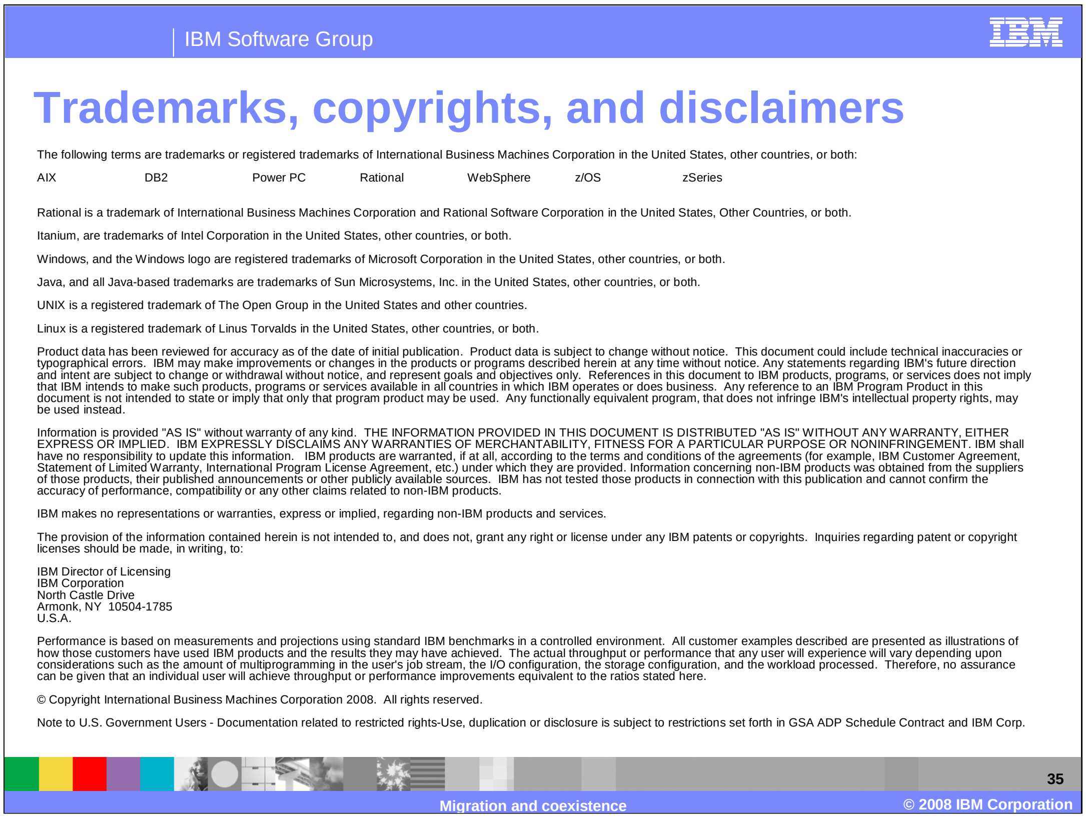
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB61_IEA_Migration.ppt

This module is also available in PDF format at: ../WMB61_IEA_Migration.pdf

You can help improve the quality of IBM Education Assistant content by providing 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 Power PC Rational WebSphere z/OS zSeries

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

Itanium, are trademarks of Intel 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.

Java, 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.

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.