



IBM Software Group

WebSphere Software

WebSphere MQ: Migration through co-existence (Distributed Platforms)

Jason Edmeades

Jason_edmeades@uk.ibm.com



**WebSphere Integration
User Group (UK)**

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.

Agenda

- **Version-to-version migration**
- **Multiple installations**
- **The primary installation**
- **Application connectivity**
- **Migration scenarios**
- **Enabling new function delivered between releases**

Agenda

- **Version-to-version migration**
- **Multiple installations**
- **The primary installation**
- **Application connectivity**
- **Migration scenarios**
- **Enabling new function delivered between releases**

“Big-bang” migration

- 1. Stop local MQ applications**
- 2. Stop ALL queue managers and listeners**
- 3. Uninstall previous MQ release’s fixpacks**
- 4. Uninstall previous MQ release**
- 5. Install new MQ release**
- 6. Restart all queue managers (strmqm)**
 - ▶ Queue manager data is migrated at this point
 - ▶ No going back now
- 7. Restart local MQ applications**

Down-time

How to make migration more flexible

- **Install additional MQ code level alongside existing installation while queue managers are still running**
 - ▶ No need to stop messaging during the installation

- **Move queue managers one at a time to the new installation**

- **Migrate and test applications one at a time**

Streamlined migration

1. Install new MQ release alongside

2. Stop local MQ applications using QM1

3. Stop queue manager QM1

4. Associate QM1 with the new installation

5. Restart QM1 (strmqm)

1. Queue manager data is migrated at this point
2. No going back now

6. Restart local MQ applications using QM1

7. Stop local MQ applications using QM2

8. Stop queue manager QM2

9. Associate QM2 with the new installation

10. Restart QM2 (strmqm)

1. Queue manager data is migrated at this point
2. No going back now

11. Restart local MQ applications using QM2

12. Uninstall previous MQ release

QM1 down-time

QM2 down-time

Maintenance, upgrade & migration - Notes

- By installing the new code level alongside an existing installation, there's no need to interrupt the flow of messages while the installation is being performed.
- Each queue manager needs to be stopped in order to be associated with the new installation.
- Migration is the process of updating queue manager data to match a newer level of code. This occurs the first time a queue manager is started with the newer level of code.
- Maintenance is the application of a fix pack, interim fix or PTF. The installation can be restored to its previous level and queue managers or applications will continue to work. No migration is required after applying maintenance.
- Upgrading is the process of taking an existing MQ installation and upgrading to a new level of code. Unless the upgrade is applying a fix (and not enabling new function), an upgrade must be followed by migration.
- Once migration has occurred, the queue manager can no longer be started by an earlier code level. Queue manager migration is not reversible (except on z/OS, with important restrictions).

Agenda

- **Version-to-version migration**
- **Multiple installations**
- **The primary installation**
- **Application connectivity**
- **Migration scenarios**
- **Enabling new function delivered between releases**

Multiple MQ installations

- **You can choose between:**
 - ▶ Simplicity of maintaining a single MQ installation
 - ▶ Flexibility of multiple MQ installations

- **Up to 128 installations of MQ on a system**
 - ▶ One of them can be v7.0.1.6 (or later)
 - Some restrictions apply while v7.0.1 is installed
 - ▶ Multiple installations of the same release are supported too
 - You can use multiple installations to help with fixpack migration

- **Single data directory for all installations**
 - ▶ All queue managers share the same name-space

Multiple MQ installations - Notes

- MQ v7.1 introduces support for multiple MQ installations on Windows, AIX, Linux, Solaris and HP-UX, but not IBM i.
- You can install multiple copies of the same code level. This is especially convenient for installation of maintenance. For example, if you want to upgrade 7.1.0.0 to 7.1.0.1, you can install a second copy of 7.1.0.0, apply the maintenance to bring it to 7.1.0.1, and then move the queue managers across to the new installation. You still have the original installation so it is a simple matter to move the queue managers back if you encounter any problems.
- All installations share a data directory. This is where you'll find mqs.ini for example.
- All installations share the same namespace for queue managers. This means that you can't create several queue managers of the same name with different installations.

So what's changed?

■ Previously:

- ▶ MQ installed into a fixed place (except on Windows)
- ▶ MQ resources had machine-scope preventing >1 installation

■ Now:

- ▶ MQ installations are relocatable
 - Each installation has a separate installation path
- ▶ MQ resources have installation-scope
 - Resource isolation so operations on one installation do not affect the others
- ▶ Queue managers are “associated” with an installation
 - You can move them, but you can't migrate data back to earlier releases

So what's changed? - Notes

- Previously, there was only one MQ installation permitted at a time. To move to multiple installations, some details of how MQ works have had to change.
- Now, MQ installations are fully relocatable. You can choose where you would like to install MQ
- Any global named resources created by MQ are now qualified with installation-specific information.
- This means that the resources created by one installation are isolated from those created by other installations. It enables things such as removing an installation of MQ while queue managers are running under another installation.

Working with multiple installations

- To work with a queue manager, you need to use the commands from its installation
 - ▶ If you get the wrong installation, you'll see:
 - **AMQ5691: Queue manager 'MYQM' is associated with a different installation (Inst1).**

- You can:
 - ▶ Use the full path to the commands
 - **`$ MQ_INSTALLATION_PATH\bin\strmqm MYQM`**
 - ▶ Set the environment variables for an installation with one of:
 - **`$ MQ_INSTALLATION_PATH/bin/setmqenv -s`**
 - **`$ setmqenv -m MYQM`**
 - **`$ setmqenv -n InstallationName`**
 - **`$ setmqenv -p MQ_INSTALLATION_PATH`**

Working with multiple installations - Notes

- If you have more than one MQ installation, you have to think about which installation's commands you're using.
- To work with a queue manager, you have to use the control commands from its associated installation. You have a choice of using the full path to the control commands or putting the commands' directory in your path ahead of any other directory containing MQ control commands.
- You might consider using a shell script or batch file to set up the environment for each MQ installation. You can use the `setmqenv` or `crtmqenv` commands to help with this.
 - ▶ `setmqenv` sets the values of the environment variables such as `PATH`, `CLASSPATH` and `LD_LIBRARY_PATH` for use with an MQ installation.
 - ▶ `crtmqenv` creates a list of the environment variables and their values for use with a particular MQ installation. You can then use this list to incorporate into a shell script or batch file.

Commands that work across installations

- In most cases, you need to use the commands from the right installation

- A few commands can work across installations:
 - ▶ **dspmq**
 - displays status of all queue managers
 - ▶ **dspmqver**
 - displays information about all installations
 - ▶ **dspmqinst**
 - displays information about all installations
 - ▶ **setmqinst**
 - can modify other installations

New control commands for multiple installations

| Command | Purpose |
|-----------|--|
| setmqm | Set the associated installation of a queue manager <code>setmqm -m QMgrName -n InstallationName</code> |
| setmqenv | Set up MQ environment <code>setmqenv -s</code> <code>setmqenv -m QMgrName</code> <code>setmqenv -n InstallationName</code> <code>setmqenv -p InstallationPath</code> |
| setmqinst | Set MQ installation properties <code>setmqinst -n InstallationName -i -x</code> |
| dspmqinst | Display MQ installation entries <code>dspmqinst</code> |

Agenda

- **Version-to-version migration**
- **Multiple installations**
- **The primary installation**
- **Application connectivity**
- **Migration scenarios**
- **Enabling new function delivered between releases**

The primary installation

- The installation to which system-wide locations refer
 - ▶ Prior to MQ v7.1, the only installation was primary
- Having a primary installation is optional, but convenient

- On UNIX and Linux:
 - ▶ Has symlinks in /usr/lib and /usr/bin
 - ▶ **No primary installation => no symlinks**
 - ▶ If you uninstall v7.0.1, and then install v7.1, it is not by default the primary
 - ▶ The primary installation must be MANUALLY CONFIGURED
 - `$ MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH`

- On Windows:
 - ▶ The first installation by default is configured as the primary
 - ▶ Global environment variables point to the primary installation
 - ▶ Some OS features require central registration of interface libraries
 - .NET monitor (transactional mode), COM/ActiveX interface classes

Where's my installation?

1. Use platform installation tools to query what's installed where

2. Use dspmqver to display the installations

- ```
$ dspmqver
Name: WebSphere MQ
Version: 7.1.0.0
...
InstName: CurrentVersion
InstPath: C:\mqm
Primary: No
```

Note there are a number (2) of other installations, use the '-i' parameter to display them.

## 3. Use dspmqinst to display the installations

## 4. On UNIX and Linux, you can list them:

- ```
cat /etc/opt/mqm/mqinst.ini
```

5. On Windows, you can query the registry:

- ```
reg.exe query "HKLM\Software\[Wow6432Node\]IBM\WebSphere MQ\Installation" /s
```

## *Installing multiple copies of the same release*

- **On Linux and Solaris only, you must ensure that each package installed has a unique name**
  - ▶ This adds an extra step to installation of multiple copies of the same release
  
- **You need to use a tool to create a unique set of packages:**
  - ▶ **\$ `crtmqpkg PACKAGE_SUFFIX`**
  - ▶ This takes the MQ installation packages and repackages them with a new name of your choice
  - ▶ Then, just install as usual

# *Migration of Windows registry information*

- MQ v7.1 on Windows uses mqs.ini and qm.ini
  
- Well, it's almost that easy 😊
  
- **If MQ v7.0.1 is still installed, MQ v7.1 uses the registry instead of mqs.ini**
  - ▶ To uninstall MQ v7.0.1 with v7.1 already installed, you must:
    - End ALL queue managers and listeners
    - Uninstall MQ v7.0.1
      - This will migrate the registry information into mqs.ini and qm.ini
    - Then, MQ v7.1 will start using mqs.ini instead



## *Migration of Windows registry information - Notes*

- On Windows prior to MQ v7.1, MQ stored its configuration information in the Windows registry, rather than using the mqs.ini and qm.ini files like the other distributed platforms.
- In MQ v7.1, we use mqs.ini and qm.ini on Windows too.
- When MQ v7.0.1 is still installed, we continue to use the registry instead of mqs.ini because MQ v7.0.1 does not know about mqs.ini. Consequently, when you uninstall MQ v7.0.1, the information in the registry is migrated into an mqs.ini file.
- This migration is automatic as part of the uninstall of MQ v7.0.1. It is necessary to end all queue managers and listeners while you uninstall MQ v7.0.1. Once the uninstallation is complete, MQ will start using mqs.ini instead.
- If for some reason the uninstall is interrupted and the automatic migration does not run, a tool is provided to migrate the information out of the registry manually.

# Agenda

- Version-to-version migration
- Multiple installations
- The primary installation
- **Application connectivity**
- Migration scenarios
- Enabling new function delivered between releases

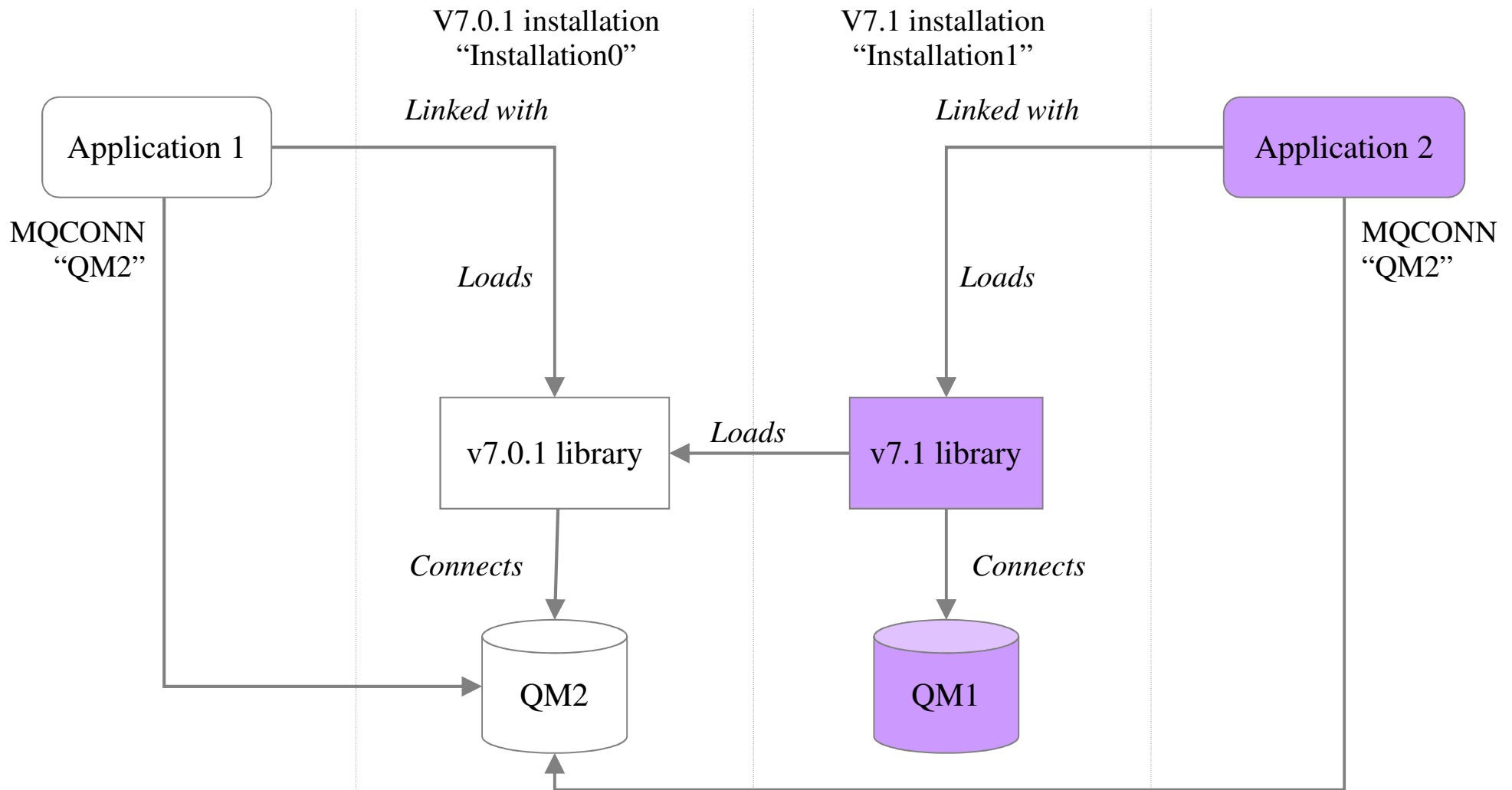
## *Multiple installations and application programs*

- **When a local application connects to a queue manager, it needs to load the libraries from the installation associated with the queue manager**
  
- **With multiple installations this introduces some complexity:**
  - ▶ When `setmqm` is used to change the installation associated with a queue manager, the libraries that need to be loaded change
  - ▶ When an application connects to multiple queue managers owned by different installations, multiple sets of libraries need to be loaded
  
- **If you link your applications to MQ v7.1 libraries, they automatically load the appropriate libraries when the application connects to a queue manager**
  - ▶ MQ v7.1 libraries can “switch” to the right libraries for the QM’s installation

## *Multiple installations and application programs - Notes*

- When a local application connects to a queue manager, it needs to load the libraries from the installation associated with the queue manager. It is vitally important that the libraries used to call into a queue manager match the queue manager's code.
- With multiple installations this introduces some complexity.
- When `setmqm` is used to change the installation associated with a queue manager, the libraries that need to be loaded change.
- When an application connects to multiple queue managers owned by different installations, multiple sets of libraries need to be loaded.
- If you link your applications to MQ v7.1 libraries, they automatically load the appropriate libraries when the application connects to a queue manager. MQ 7.1 libraries understand about multiple MQ installations and automatically load the correct installation's libraries when the application connects to a queue manager.
- This means that as long as the application program loads an MQ v7.1 library, it will be able to load the correct installation's libraries automatically.

## Application connectivity and multiple installations



## ***Loading MQ libraries in a multi-version environment***

- **How libraries are located depends on your environment**
  - ▶ If MQ v7.1 is installed in the default location, existing applications continue to work as before
  - ▶ Otherwise, you may need to rebuild the application or change configuration
  
- **On Windows, libraries are searched for in this order:**
  - ▶ The application's directory
  - ▶ The current directory
  - ▶ The global and the user's PATH variables
  
- **On other platforms, libraries are searched for in this order:**
  - ▶ LD\_LIBRARY\_PATH (or LIBPATH/SHLIB\_PATH)
  - ▶ An embedded search path (RPath)
  - ▶ The default library path

# Options for loading libraries

| Platform       | Option                                              | Benefits                                                                                                                                                                                    | Drawbacks                                                                                                                                                                        |
|----------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UNIX and Linux | Set/change the embedded runtime search path (RPath) | <ul style="list-style-type: none"> <li>The path is explicit in the way the application is built</li> </ul>                                                                                  | <ul style="list-style-type: none"> <li>You need to recompile/link</li> <li>If you move MQ, you must change RPath</li> </ul>                                                      |
| UNIX and Linux | Set LD_LIBRARY_PATH or equivalent using setmqenv    | <ul style="list-style-type: none"> <li>No changes to existing applications</li> <li>Overrides RPath</li> <li>Easy to change if you move MQ</li> </ul>                                       | <ul style="list-style-type: none"> <li>Depends on environment vars</li> <li>Possible impacts on other applications</li> </ul>                                                    |
| Windows        | Set PATH using setmqenv                             | <ul style="list-style-type: none"> <li>No changes to existing applications</li> <li>Easy to change if you move MQ</li> </ul>                                                                | <ul style="list-style-type: none"> <li>Depends on environment vars</li> <li>Possible impacts on other applications</li> </ul>                                                    |
| All            | Set the primary installation to v7.1 or later       | <ul style="list-style-type: none"> <li>No changes to existing applications</li> <li>Easy to change the primary installation</li> <li>Similar behavior to previous versions of MQ</li> </ul> | <ul style="list-style-type: none"> <li>While v7.0.1 is installed, you cannot make v7.1 primary</li> <li>UNIX and Linux: Relies on /usr/lib in the default search path</li> </ul> |



# Agenda

- Version-to-version migration
- Multiple installations
- The primary installation
- Application connectivity
- **Migration scenarios**
- Enabling new function delivered between releases

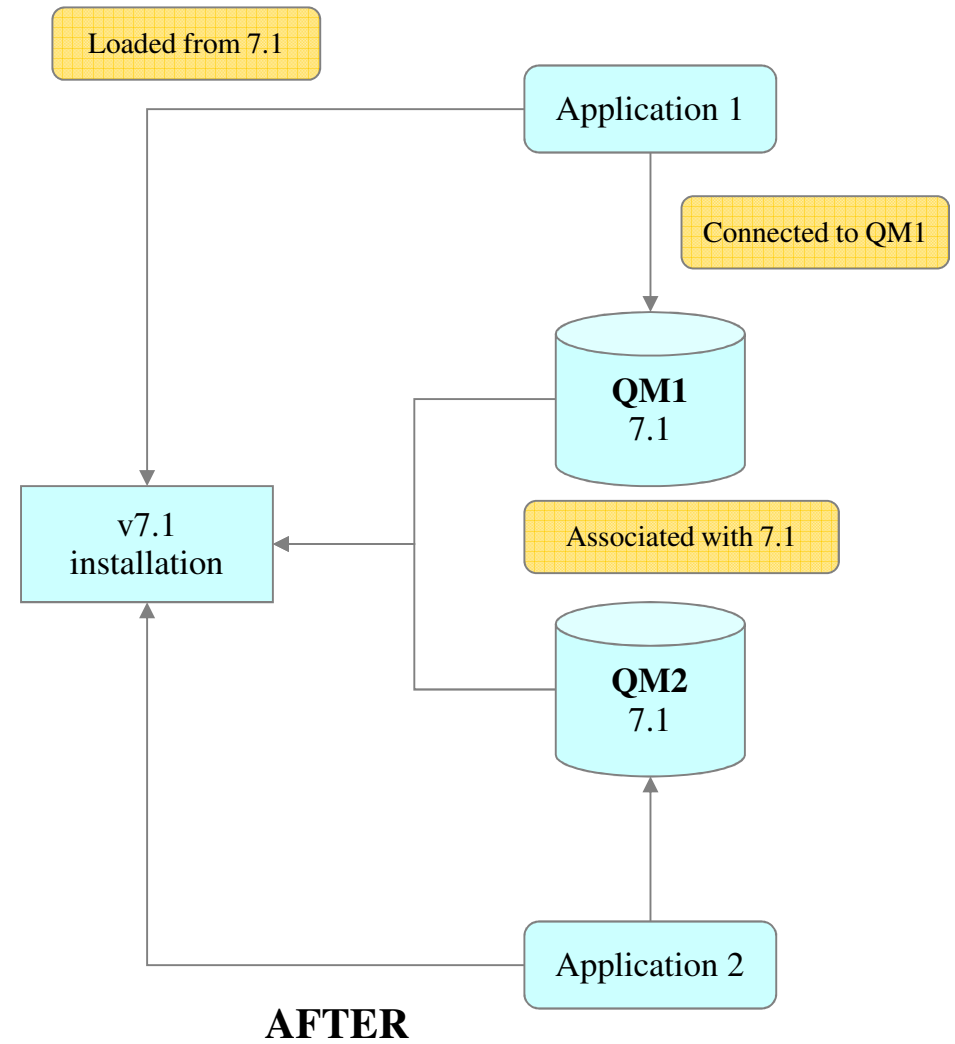
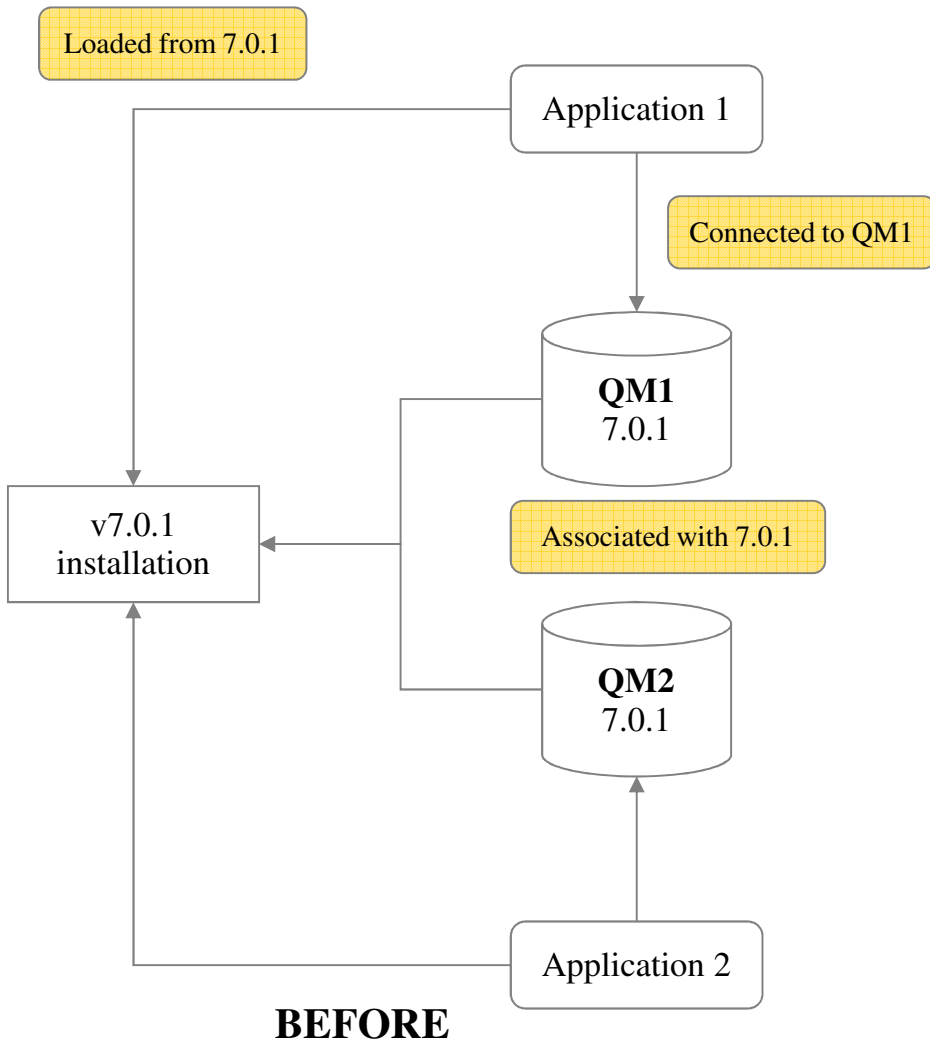
# Migration scenarios

- **Three example scenarios for migrating v7.0.1 to v7.1**
  - ▶ Single-stage migration
    - Upgrade v7.0.1 to v7.1 in the same, default location
  - ▶ Side-by-side migration
    - Install v7.1 alongside v7.0.1 in an alternate location
    - Migrate all queue managers and applications at the same time
  - ▶ Multi-stage migration
    - Install v7.1 alongside v7.0.1 in an alternate location
    - Migrate queue managers and applications one-by-one
    - Application migration more complex
    - Installations run in parallel
      - MQ environment needs to be set explicitly

# Migration scenarios – Notes

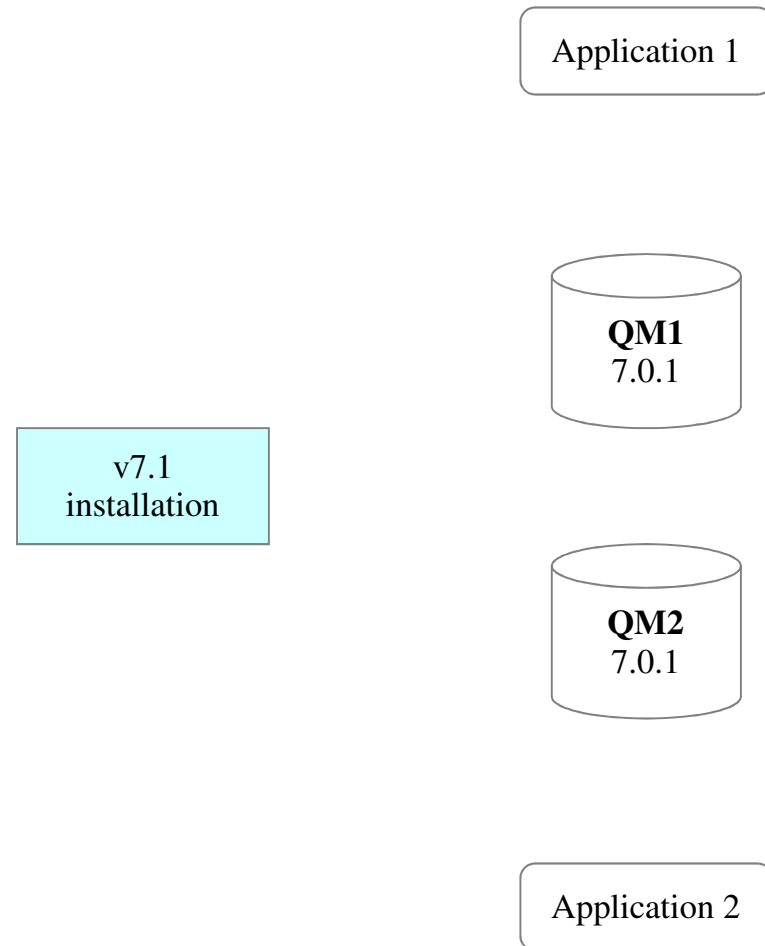
- Single-stage migration is the term used to describe replacing the only installation of WebSphere® MQ on a server, with a later release. Until version 7.1, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.
- Side-by-side migration is the term used to describe installing a new version of WebSphere® MQ alongside an older version on the same server. Queue managers remain running during the installation and verification of the new version of WebSphere MQ. They remain associated with the older version of WebSphere MQ. When you decide to migrate queue managers to the new version of WebSphere MQ, you stop all queue managers, migrate them all to the new version, and uninstall the old version of WebSphere MQ.
- The side-by-side migration scenario is less flexible than the multi-stage migration. The advantage the side-by-side scenario has over the single-stage scenario is that you can install and verify the version 7.1 installation on the server before switching over to it. With the side-by-side approach, you assign a version 7.1 installation to be primary, whereas with the multi-stage approach, you cannot do so until you uninstall version 7.0.1. With a version 7.1 primary installation, many applications restart without reconfiguring their environment and WebSphere MQ commands work without providing a local search path.
- Multi-stage migration is the term used to describe running a new version of WebSphere® MQ alongside an older version on the same server. After installing the new version alongside the old, you can create new queue managers to verify the new installation, and develop new applications. At the same time, you can migrate queue managers and their associated applications from the old version to the new. By migrating queue managers and applications one-by-one, you can reduce the peak workload on staff managing the migration.

# Scenario 1 – Single-stage migration



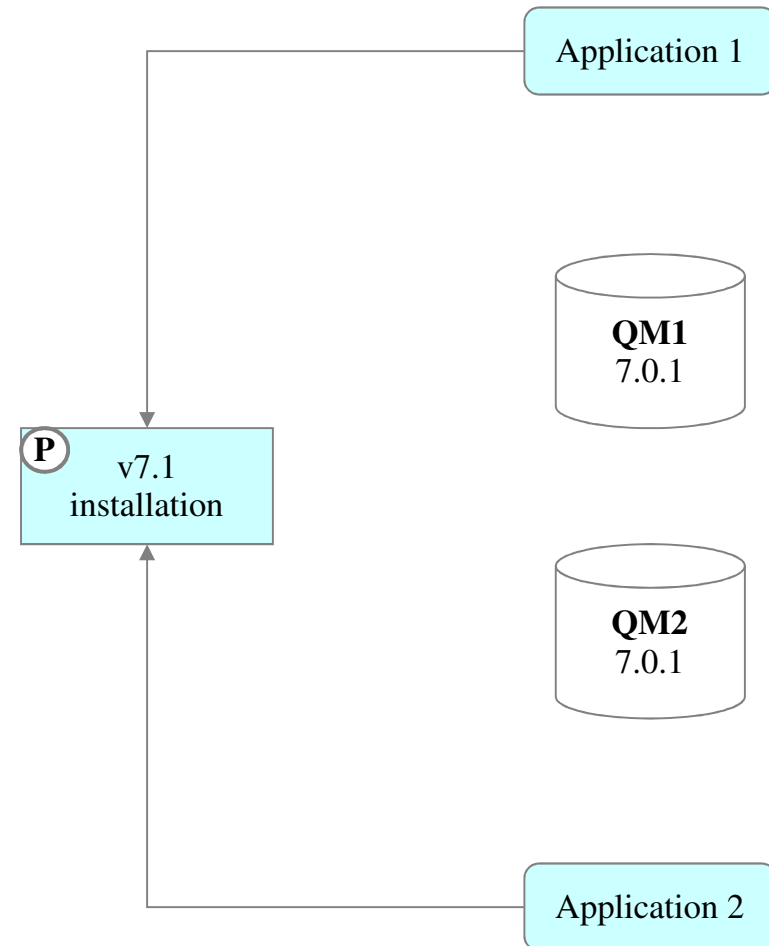
# Scenario 1 – Single-stage migration

- Upgrade v7.0.1 to v7.1 in the same installation directory
  - ▶ Stop all local MQ applications
  - ▶ Stop all queue managers and listeners
  - ▶ Upgrade v7.0.1 to v7.1
    - Upgrade in place, or
    - Uninstall v7.0.1, install v7.1
  - ▶ Make sure not to delete the queue managers
  - ▶ Because the same directory is being used, application migration is simplified



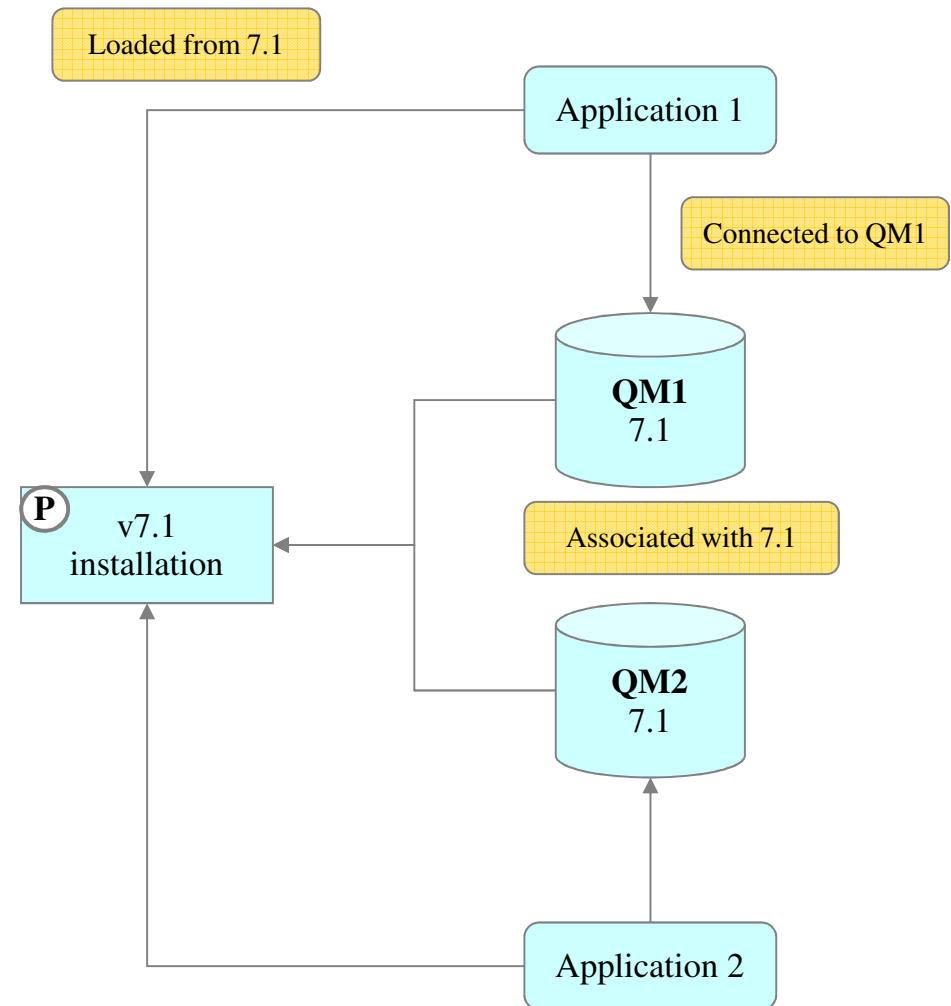
# Scenario 1 – Single-stage migration

- Make the v7.1 installation primary
  - ▶ `setmqinst -i -n Inst1`
  - ▶ Avoids specifying a search path on MQ commands



# Scenario 1 – Single-stage migration

- Start the queue managers and applications
  - ▶ Optionally, associate the queue managers with the installation
    - `setmqm -m QM1 -n Inst1`
    - `setmqm -m QM2 -n Inst1`
  - ▶ Start the queue managers to upgrade them to v7.1
    - `strmqm QM1`
    - `strmqm QM2`
  - ▶ Start the applications
    - v7.1 library checks the installation and can load the appropriate library

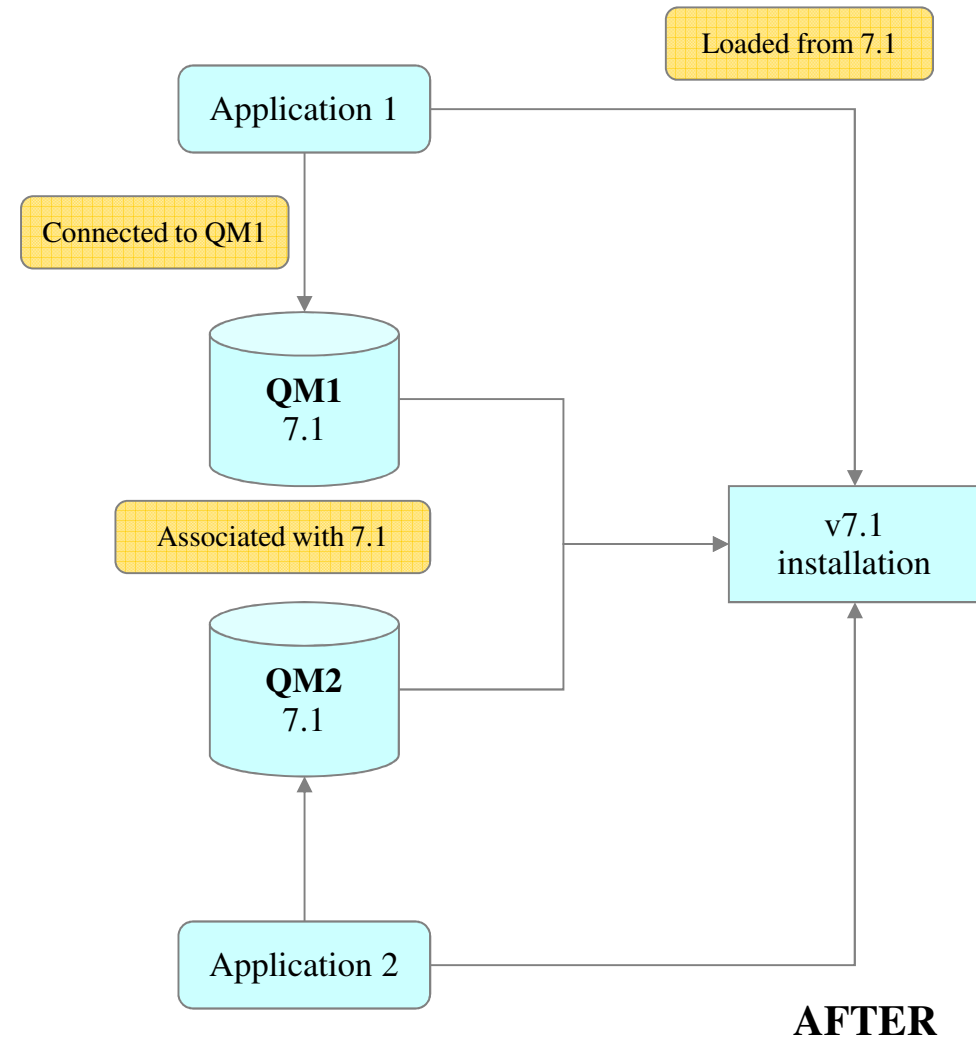
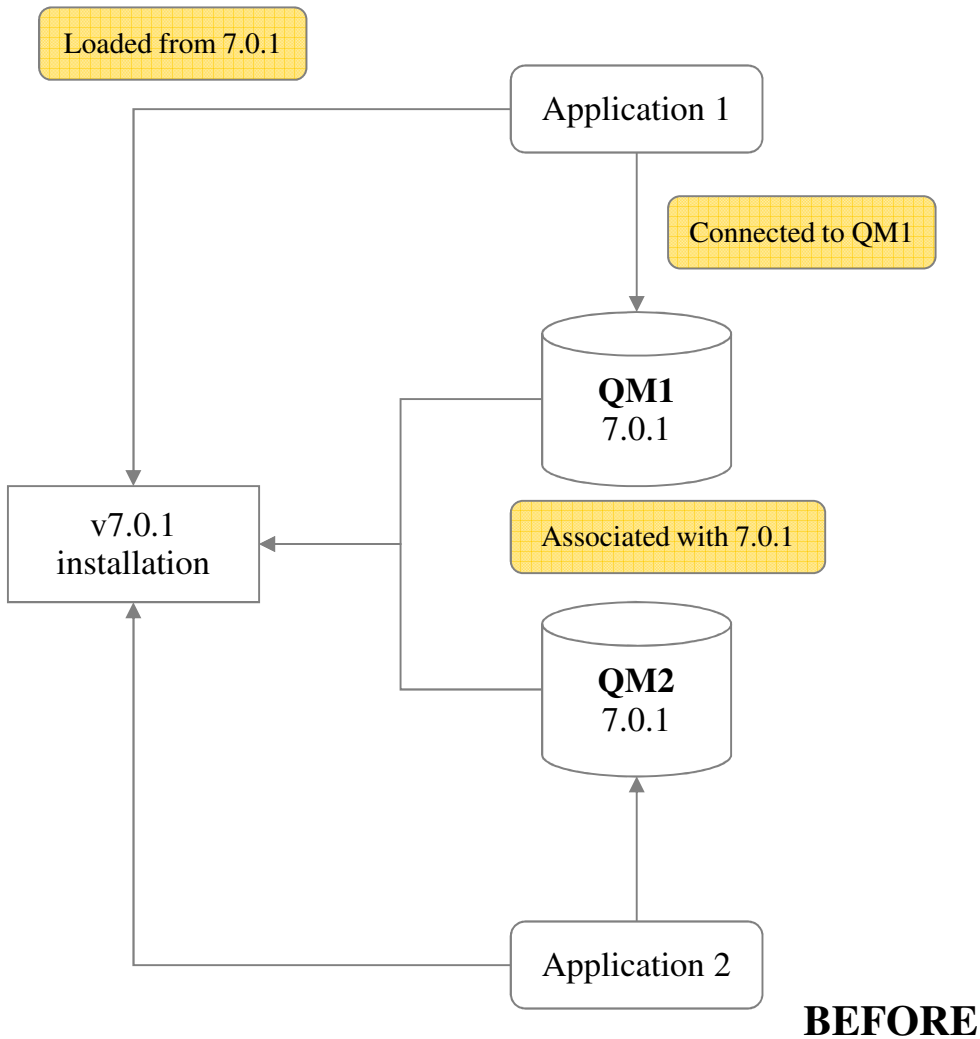




## ***Scenario 1 – Single-stage migration – Notes***

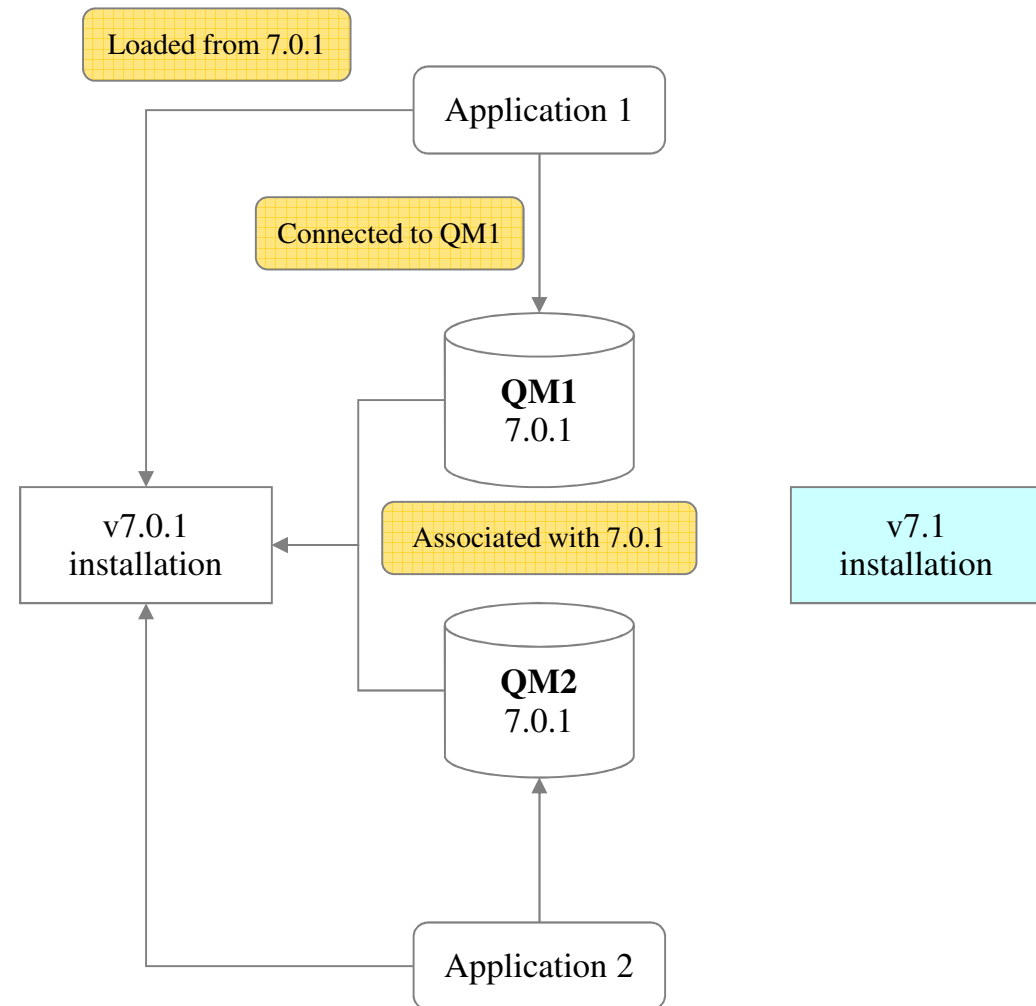
- The setmqm step is optional only in the case where migration is from v7.0.1 to a later release. In this case, the strmqm command automatically associates the queue manager with its own installation.
- If you're doing any other migration, it will be necessary to use setmqm to associate with the new installation manually.

# Scenario 2 – Side-by-side migration



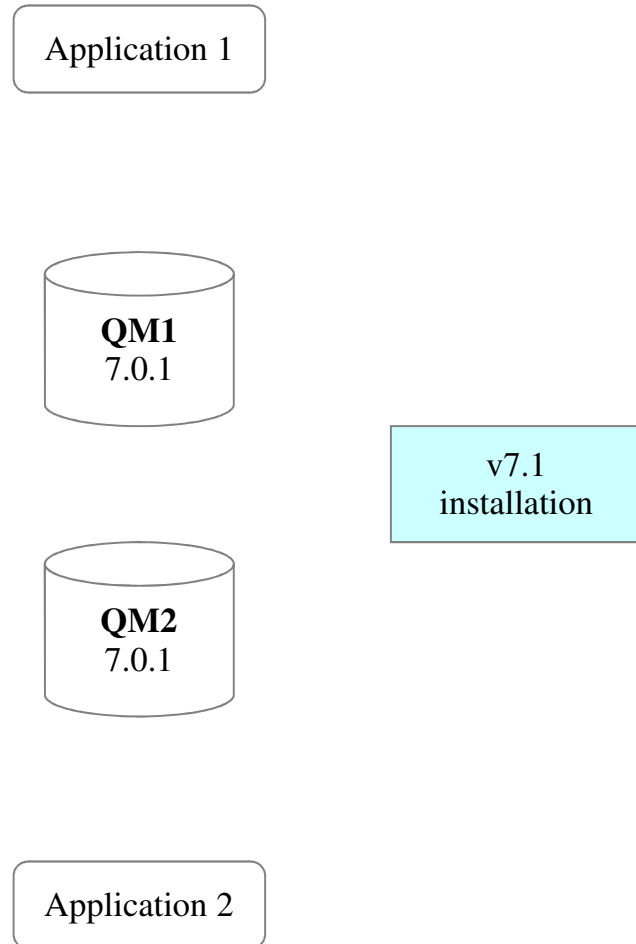
## Scenario 2 – Side-by-side migration

- Install v7.1 in a different installation directory to v7.0.1
- Verify the installation



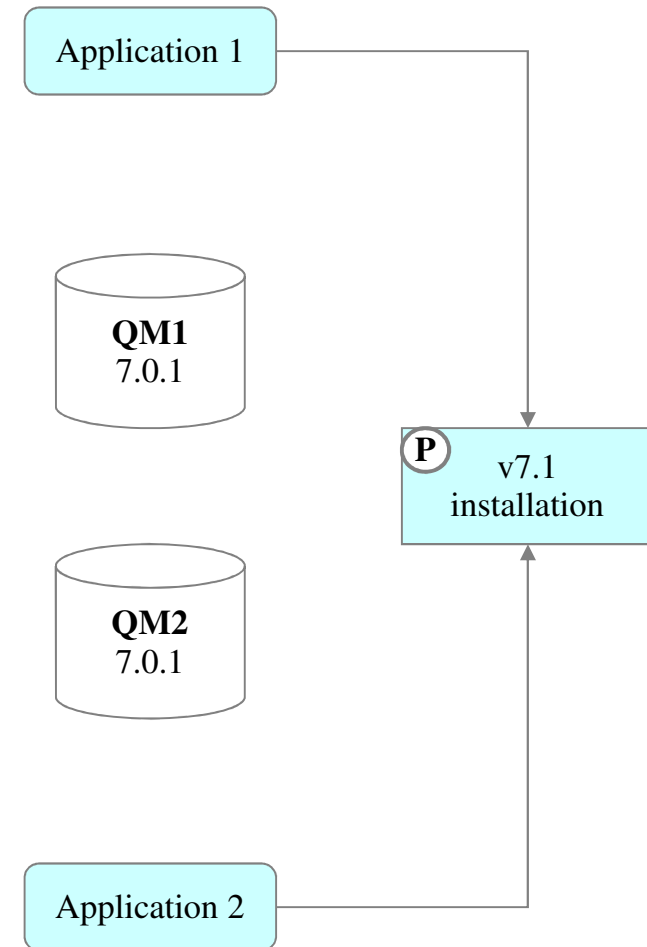
## Scenario 2 – Side-by-side migration

- **Uninstall v7.0.1**
  - ▶ Stop all local MQ applications
  - ▶ Stop all queue managers and listeners
  - ▶ Make sure not to delete the queue managers
  
- **When upgrading from later releases, it will no longer be necessary to stop ALL queue managers at this point**



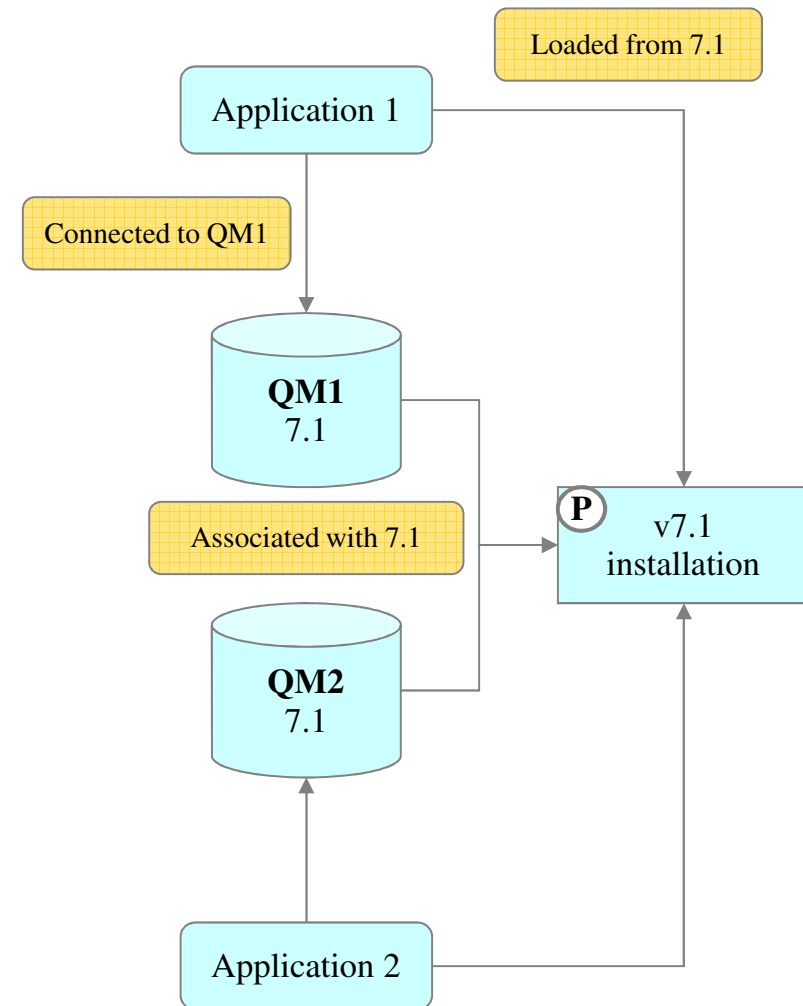
## Scenario 2 – Side-by-side migration

- **Make the v7.1 installation primary**
  - ▶ `setmqinst -i -n Inst1`
  - ▶ Avoids specifying a search path on MQ commands
  - ▶ On Linux and UNIX, applications linked to `/usr/lib` find the MQ libraries
  - ▶ With a primary v7.1 installation, an application can connect to ANY queue manager associated with ANY installation

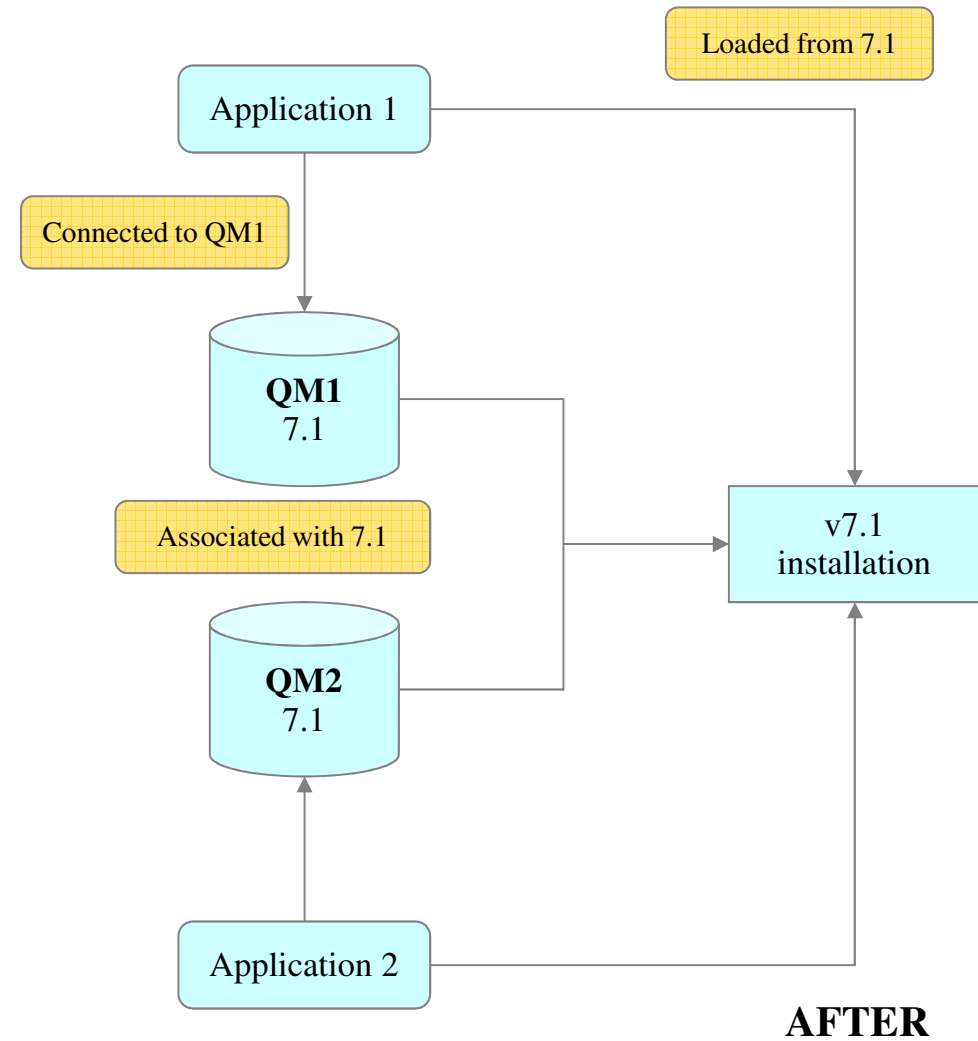
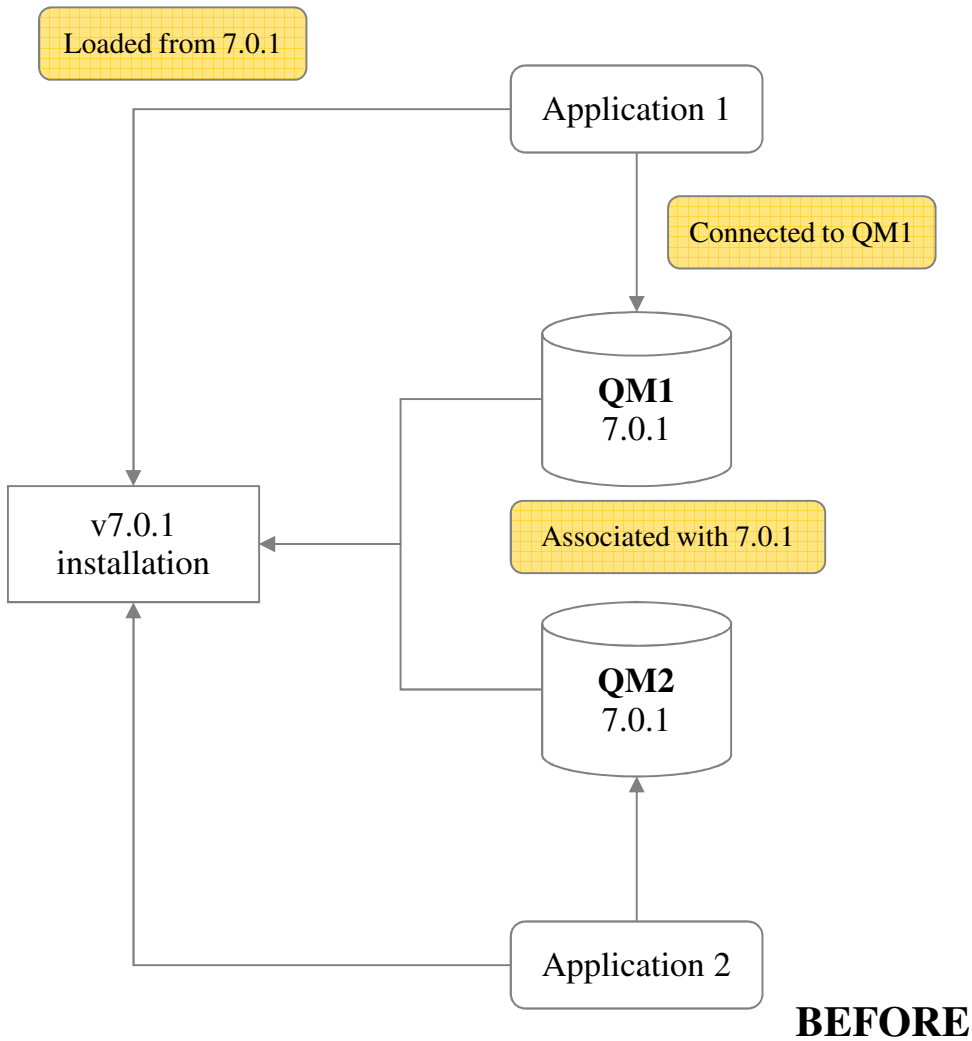


## Scenario 2 – Side-by-side migration

- Start the queue managers and applications
  - ▶ Optionally, associate the queue managers with the installation
    - `setmqm -m QM1 -n Inst1`
    - `setmqm -m QM2 -n Inst1`
  - ▶ Start the queue managers to upgrade them to v7.1
    - `strmqm QM1`
    - `strmqm QM2`
  - ▶ Start the applications
    - v7.1 library checks the installation and can load the appropriate library

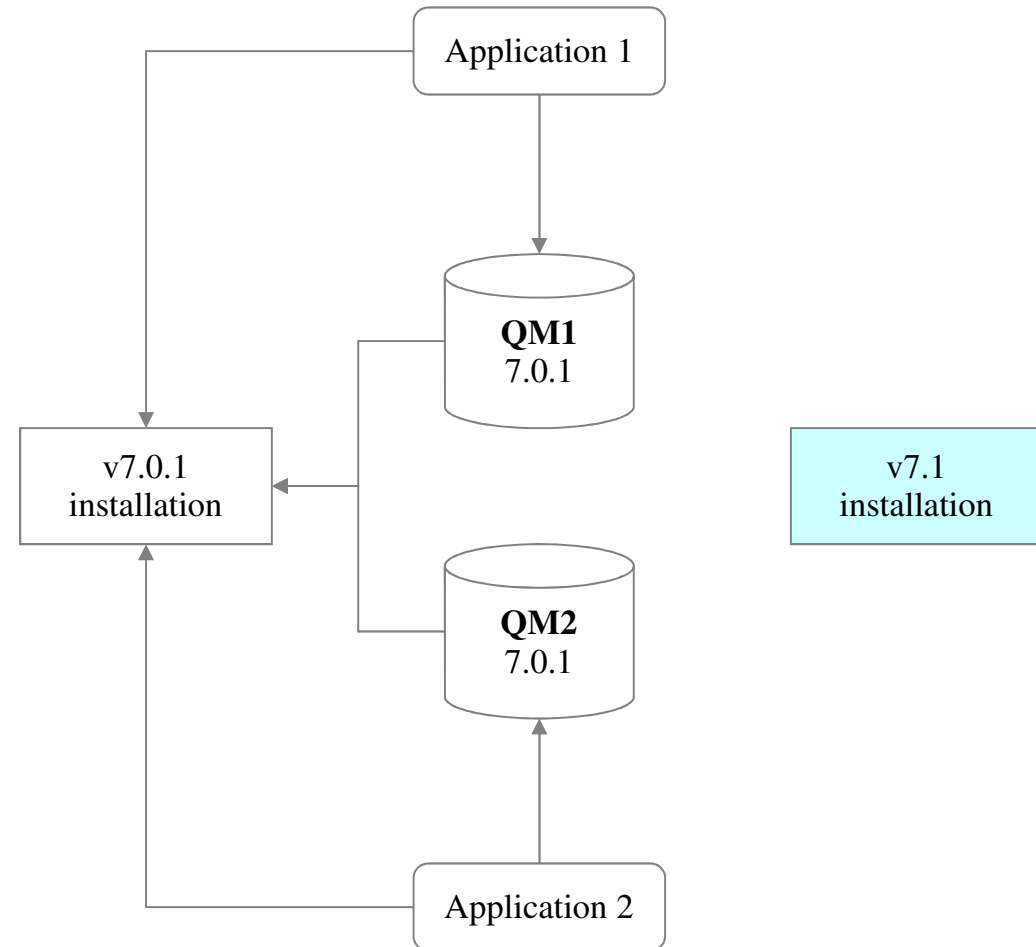


# Scenario 3 – Multi-stage migration



## Scenario 3 – Multi-stage migration

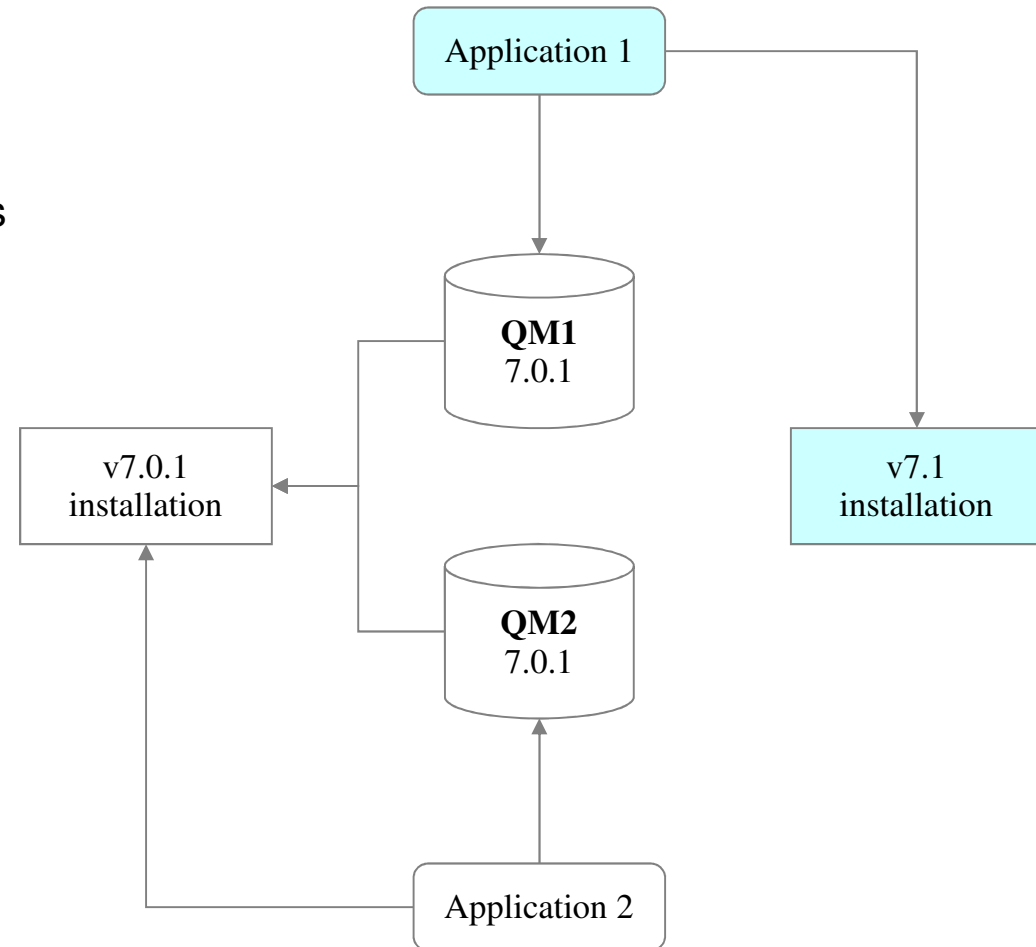
- Install v7.1 in a different installation directory to v7.0.1
- Verify the installation





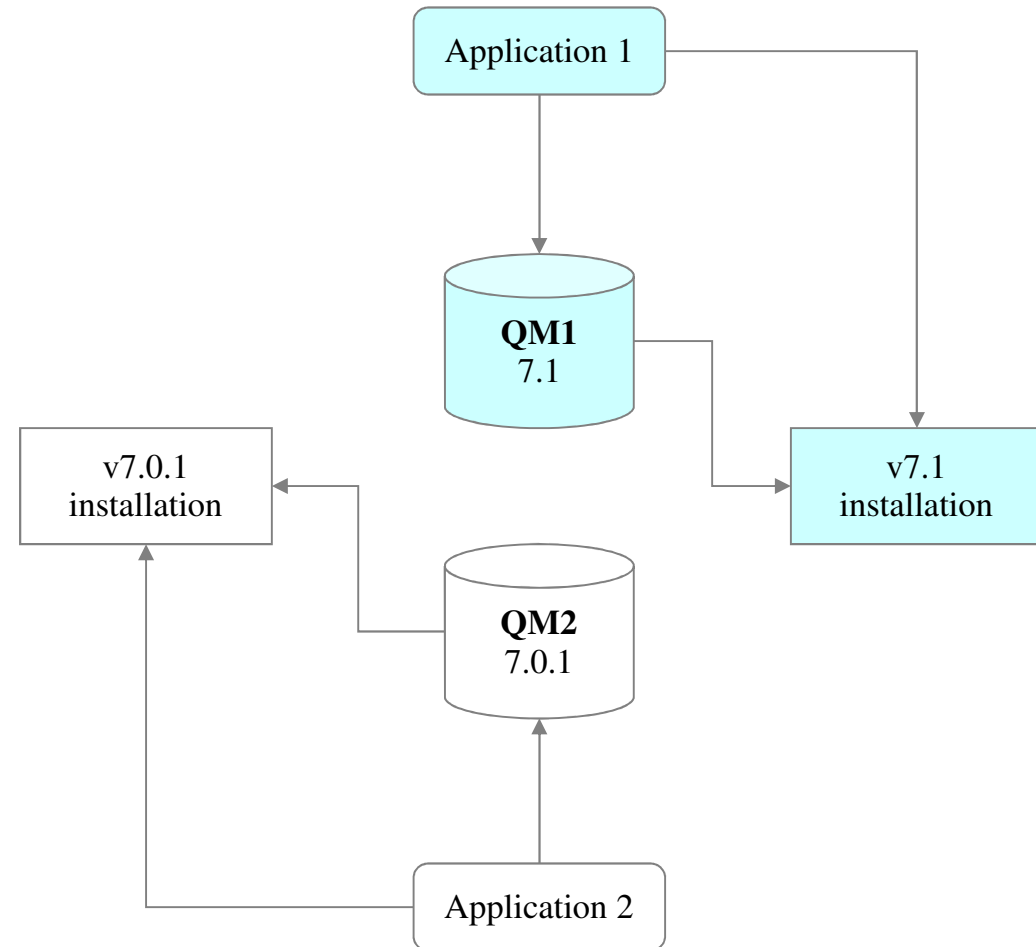
## Scenario 3 – Multi-stage migration

- Migrate an application to load the v7.1 MQ library
  - ▶ Run `setmqenv` to modify the local search path
    - `INST1_PATH/setmqenv -s`
  - ▶ Modify the global search path for libraries
  - ▶ Relink applications with an additional runtime load path



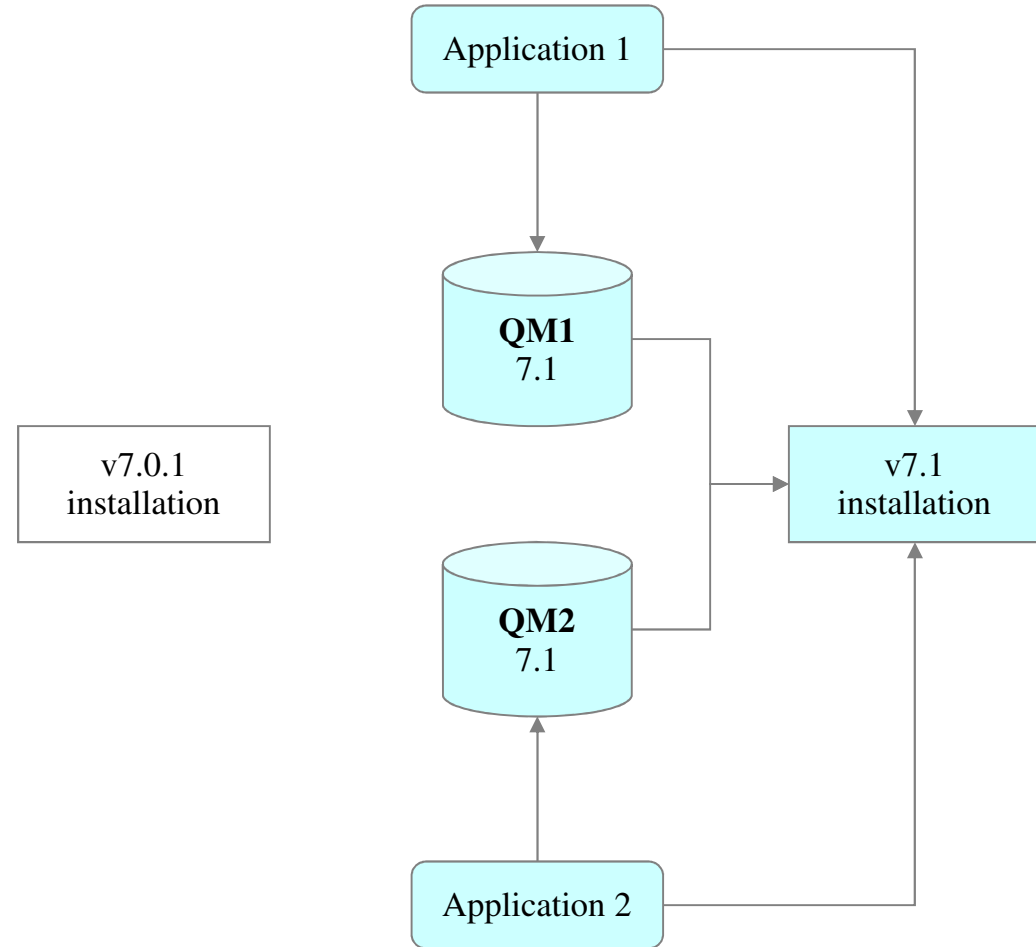
## Scenario 3 – Multi-stage migration

- Restart QM1 and the applications that connect to it
  - ▶ Set the MQ environment
    - `INST1_PATH/setmqenv -s`
  - ▶ Optionally, associate the queue manager with the installation
    - `setmqm -m QM1 -n Inst1`
  - ▶ Start the queue manager to upgrade it to v7.1
    - `strmqm QM1`
  - ▶ Start the applications
    - v7.1 library checks the installation and can load the appropriate library



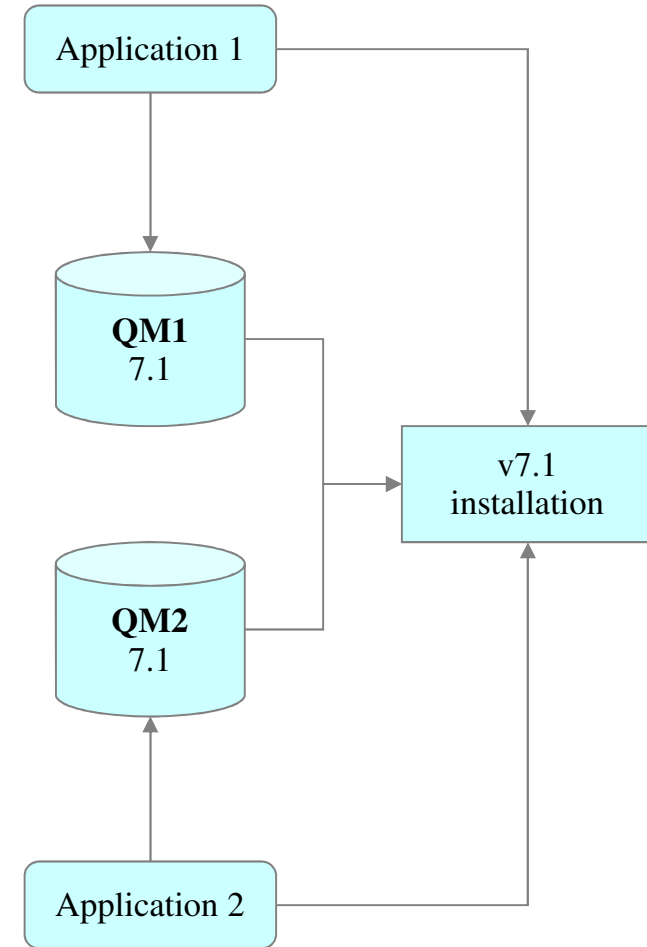
# Scenario 3 – Multi-stage migration

- Repeat migration for all applications and queue managers



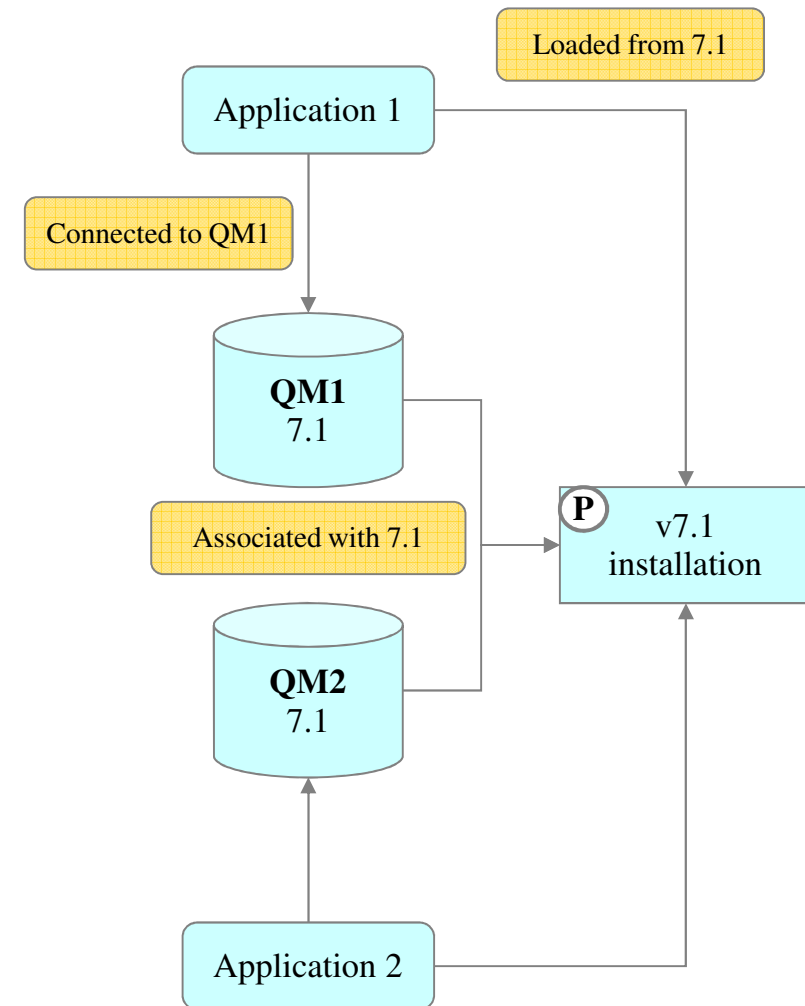
# Scenario 3 – Multi-stage migration

- Uninstall v7.0.1



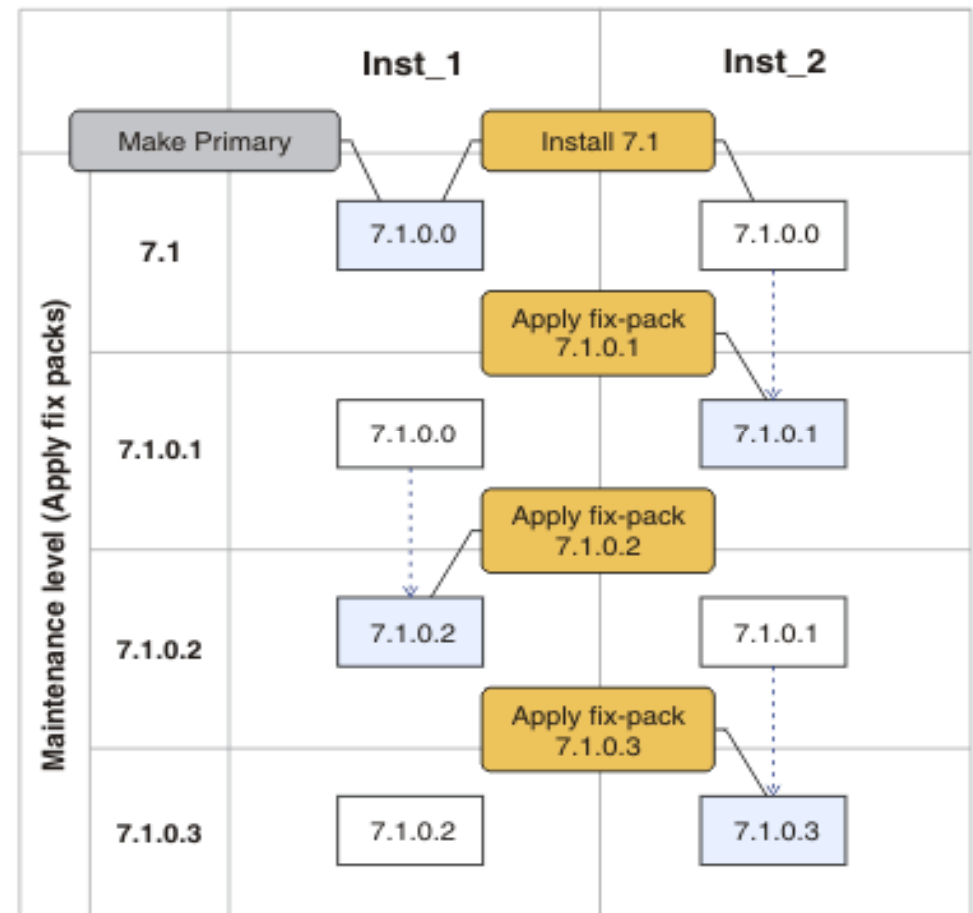
## Scenario 3 – Multi-stage migration

- Optionally, make the v7.1 installation primary
  - ▶ `setmqinst -i -n Inst1`
  - ▶ Avoids specifying a search path on MQ commands



## Applying Maintenance (Based on Scenario 3)

- Can use multiple installations to provide rolling fix pack levels
  - ▶ Primaryness
  - ▶ Library load paths
  - ▶ Do NOT rely on switching
  
- Swap between installations
  - ▶ Primaryness
  - ▶ Library load paths
  - ▶ Do NOT rely on switching
  
- Makes rollback immediate
  - ▶ Setmqm back again



# Agenda

- Version-to-version migration
- Multiple installations
- The primary installation
- Application connectivity
- Migration scenarios
- **Enabling new function delivered between releases**

## ***Enabling new function delivered between releases***

- **Occasionally, IBM releases new function in modification releases and fix packs**
  - ▶ e.g. Multi-instance QM in 7.0.1, preconnect exit in 7.0.1.4
  
- **We do this to satisfy important requirements between releases**
  
- **Any new function introduced between releases must be explicitly enabled if required**
  - ▶ If new function is not enabled, the fix pack just behaves like a bunch of fixes
  - ▶ Protects customers against accidental use of new function and changes in behavior



# New function switch

- **Delivery of significant function between releases accompanied by a new command level**
  - ▶ For example, v7.1 is CMDLEVEL(710)
  - ▶ If we delivered a new function, it would be CMDLEVEL(>710)
  - ▶ First use was CERTVPOL (711) in 7.1.0.2
    - [https://www.ibm.com/developerworks/mydeveloperworks/blogs/aimsupport/entry/using\\_certvpol\\_to\\_specify\\_rfc\\_5280\\_certificate\\_validation\\_for\\_queue\\_managers?lang=en](https://www.ibm.com/developerworks/mydeveloperworks/blogs/aimsupport/entry/using_certvpol_to_specify_rfc_5280_certificate_validation_for_queue_managers?lang=en)
  
- **New function is enabled by increasing the command level of the queue manager**
  - ▶ `strmqm -e CMDLEVEL=711 QmgrName`
    - The queue manager starts up, enables the function, migrates the QM's data, and stops
  - ▶ The queue manager can now only be started by an installation that supports this command level (minimum code level)
  
- **If you are going to enable new function, try it out on a test system first, regression test your applications and then move into production**
  - ▶ Treat it like taking a new version
  - ▶ But if you don't want the new function, you can safely just ignore it

# Example of enabling new command levels

| MQ 7.1.0.0<br>Max CMDLEVEL(710)                      | MQ 7.1.0.x<br>Max CMDLEVEL(711)                                          | MQ 7.1.0.y<br>Max CMDLEVEL(712)                                          | MQ "Next"<br>Max CMDLEVEL(2000)                       |
|------------------------------------------------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------|-------------------------------------------------------|
| crtmqm<br>⇒CMDLEVEL(710)<br>⇒Min code level: 7.1.0.0 | Install & enable up to 711<br>⇒CMDLEVEL(711)<br>⇒Min code level: 7.1.0.x | Install & enable up to 712<br>⇒CMDLEVEL(712)<br>⇒Min code level: 7.1.0.y | Install<br>⇒CMDLEVEL(2000)<br>⇒Min code level: "Next" |
|                                                      |                                                                          | Install only<br>⇒CMDLEVEL(711)<br>⇒Min code level: 7.1.0.x               | Install<br>⇒CMDLEVEL(2000)<br>⇒Min code level: "Next" |
|                                                      | Install only<br>⇒CMDLEVEL(710)<br>⇒Min code level: 7.1.0.0               | Install & enable up to 712<br>⇒CMDLEVEL(712)<br>⇒Min code level: 7.1.0.y | Install<br>⇒CMDLEVEL(2000)<br>⇒Min code level: "Next" |
|                                                      |                                                                          | Install & enable up to 711<br>⇒CMDLEVEL(711)<br>⇒Min code level: 7.1.0.x | Install<br>⇒CMDLEVEL(2000)<br>⇒Min code level: "Next" |
|                                                      |                                                                          | Install only<br>⇒CMDLEVEL(710)<br>⇒Min code level: 7.1.0.0               | Install<br>⇒CMDLEVEL(2000)<br>⇒Min code level: "Next" |
|                                                      |                                                                          | Install only<br>⇒CMDLEVEL(710)<br>⇒Min code level: 7.1.0.0               | Install<br>⇒CMDLEVEL(2000)<br>⇒Min code level: "Next" |

The details here are totally fictitious and merely illustrative of the behavior of CMDLEVEL.

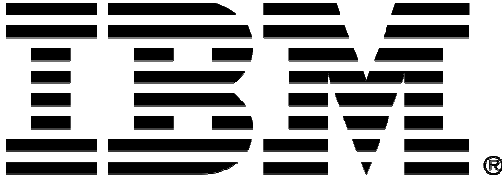
# Summary

- **The options for version-to-version migration are greatly increased by support for multiple installations**
  - ▶ No more big-bang migrations
  - ▶ Side-by-side installation, particularly helpful for fix packs
  - ▶ Granular application migration
  - ▶ Relocatable installations

# Questions?



---



IBM®

The image shows the classic IBM logo, which consists of the letters "IBM" in a bold, sans-serif font. Each letter is formed by eight horizontal black bars of equal thickness, stacked vertically. The letters are positioned in the center of the page. To the right of the "M" is a small registered trademark symbol (®). The background is plain white.

## Legal Disclaimer

- © IBM Corporation 2013. All Rights Reserved.
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:  
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:  
All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to <http://www.ibm.com/legal/copytrade.shtml> for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:  
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:  
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:  
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:  
Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:  
UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:  
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.