Migrating between releases - OPMODE in MQ V9 LTS for z/OS

[gwydiontudur](javascript:;" \o "gwydiontudur) |Apr 7 2017 Updated

When migrating between MQ releases on z/OS, it’s possible to fall back to the previous version of MQ. So you may, for example, migrate from MQ V8 to V9, back to V8, then forward to V9, before deciding to stay at V9. There are some new capabilities in each new release of MQ, which if used, would prevent the queue manager from falling back to the previous version. For example, in MQ V8, 8 byte log RBA and host names in channel authentication rules.

To prevent the accidental use of these new capabilities you have to explicitly set a switch to indicate that you do not want to go back to an earlier release, and to allow these new capabilities to be used. You set this switch with the OPMODE parameter.

The idea is that you migrate your queue manager to the new release of MQ, perform enough testing to reassure yourself that you are happy to stay at that release, then enable the restricted new capabilities using OPMODE, which prevents the queue manager from being started at an earlier release.

As the behaviour of OPMODE has changed slightly in MQ V9, I thought I’d take this opportunity to explain how it works in a bit more detail, and what’s changed in MQ V9.

This article deals only the behaviour of OPMODE in the Long Term Service (LTS) release of MQ V9, that is, MQ V9.0.0. The behaviour of OPMODE for the Continuous Delivery releases of MQ V9 is different, and I hope to cover that in a future blog post.

**Setting OPMODE**

Setting OPMODE requires recompiling the system parameter module (ZPARM), and restarting the queue manager.

You set OPMODE for a queue manager with the CSQ6SYSP macro when compiling the ZPARM. The format of the OPMODE parameter of CSQ6SYSP is:

OPMODE=(*mode*,*verificationLevel*)

*mode* can be set to either COMPAT (the default) or NEWFUNC. If COMPAT is specified, you cannot use the restricted new capabilities. If NEWFUNC is specified, the restricted new capabilities will be available, but you will not be able to start the queue manager at an earlier release of MQ.

*verificationLevel* should be set to the VRM (version, release, modification) code for the level of MQ that you are working with. For example, this would be 900 for MQ V9.0.0. This is used as a check to ensure that you do not accidentally set NEWFUNC for a queue manager by resubmitting JCL used for a previous release of MQ. You will get a compile time error if *verificationLevel* does not match the version of CSQ6SYSP that you are running.

The value of OPMODE when the ZPARM was compiled is displayed in message CSQY101I during queue manager startup.

[image](https://www.ibm.com/developerworks/community/blogs/messaging/resource/BLOGS_UPLOADED_IMAGES/csqy101.PNG)

**Which capabilities are restricted by OPMODE at MQ V9?**

Previous releases of MQ have all introduced new capabilities that are restricted by OPMODE. MQ V9 is different. There are no new capabilities added in MQ V9 that are restricted by OPMODE.

This does not mean that OPMODE has no effect in MQ V9, because capabilities that were restricted by OPMODE in MQ V8, are still restricted in MQ V9. This is a change from previous releases of MQ, where typically a queue manager migrated to a new version of MQ would be able to use all the capabilities of earlier releases of MQ, regardless of the value of OPMODE in its ZPARM.

This means that a MQ V9 queue manager started with OPMODE=(COMPAT,900) **cannot** use the V8 restricted capabilities such as 8 byte log RBA and host names in channel authentication rules. A MQ V9 queue manager started with OPMODE=(NEWFUNC,900) or OPMODE=(NEWFUNC,800) **can** use the V8 restricted capabilities.

**When should I recompile the ZPARM?**

You do not need to recompile the ZPARM when migrating a queue manager to MQ V9 from a previous release. An MQ V9 queue manager will start with ZPARM built with previous releases of MQ. You should only recompile the ZPARM with OPMODE=(NEWFUNC,900) **when you are certain that you will not need to revert to running at a previous release of MQ.**

There are three scenarios to consider.

**New MQ V9 queue manager**

By default, the ZPARM for a new MQ V9 queue manager will be built with OPMODE=(COMPAT,900). This means that capabilities restricted by OPMODE in MQ V8 will not be available. To allow all MQ V8 and V9 capabilities to be used, you need to recompile the ZPARM with OPMODE=(NEWFUNC,900).

This table shows how OPMODE affects which capabilities are available for a new MQ V9 queue manager.

|  |  |
| --- | --- |
| **OPMODE value in ZPARM** | **Available functions** |
| COMPAT,900 | All V7.1 and earlier functions |
| NEWFUNC,900 | All functions |

**Migrating from MQ V7.1 to MQ V9**

A queue manager migrated to MQ V9 from MQ V7.1 will start in compatibility mode with only capabilities restricted in MQ V7.1 available. You do not need to recompile the ZPARM as part of the migration process. If you need to recompile the ZPARM at V9 for some reason, setting OPMODE=(COMPAT,900) will still allow the queue manager to be started at MQ V7.1.

Once you are sure that you will not need to fall back to MQ V7.1, you can recompile the ZPARM with OPMODE=(NEWFUNC,900) to allow all new capabilities to be enabled. This operation will prevent the queue manager from being started at a release earlier than MQ V9.

This table shows how OPMODE affects which capabilities are available for a queue manager migrated from MQ V7.1 to MQ V9.

|  |  |  |
| --- | --- | --- |
| **OPMODE value in ZPARM** | **Available functions** | **Release queue manager can fall back to** |
| COMPAT,710 or NEWFUNC,710 or COMPAT,900 | All V7.1 and earlier functions | V7.1 |
| NEWFUNC,900 | All functions | None – OPMODE prevents backward migration |

**Migrating from MQ V8 to MQ V9**

You should rebuild the ZPARM after you have successfully migrated to MQ V9, and will not go back to MQ V8, rather than as part of migration process.

The capabilities available when the queue manager is started at MQ V9 depends on the value of OPMODE in the ZPARM, and will be the same as the capabilities available before the queue manager was migrated.

If the ZPARM was built at MQ V8 with OPMODE=(COMPAT,800):

* The queue manager will start at MQ V9 with only MQ V7.1 capabilities available.
* If you need to recompile the ZPARM for some reason, setting OPMODE=(COMPAT,900) will still allow the queue manager to be started at MQ V8. This will not allow MQ V8 capabilities to be used.
* Once you are sure that you will not need to fall back to MQ V8, you can recompile the ZPARM with OPMODE=(NEWFUNC,900) to allow all new capabilities to be enabled.

If the ZPARM was built at MQ V8 with OPMODE=(NEWFUNC,800):

* The queue manager will start at MQ V9 with all capabilities available.
* You should not recompile the ZPARM with OPMODE=(COMPAT,900) as that will prevent MQ V8 capabilities from being used.
* If you need to recompile the ZPARM to change some other parameter, while still retaining the ability to fall back to MQ V8, specify OPMODE=(NEWFUNC,800). You will get a return code of 4 from CSQ6SYSP to warn that the verification level is different to the release of MQ, but the ZPARM will be compiled successfully.
* Plan to recompile the ZPARM with OPMODE=(NEWFUNC,900) as soon as you are sure that you will not need to fall back to MQ V8.

This table shows how OPMODE affects which capabilities are available for a queue manager migrated from MQ V8 to MQ V9.

|  |  |  |
| --- | --- | --- |
| **OPMODE value in ZPARM** | **Available functions** | **Release queue manager can fall back to** |
| COMPAT,800 or COMPAT,900 | All V7.1 and earlier functions | V8 |
| NEWFUNC,800 | All functions | V8 |
| NEWFUNC,900 | All functions | None – OPMODE prevents backward migration |

**Summary**

In summary, the main points to keep in mind when working with MQ V9 LTS is that restricted capabilities added in MQ V8 are still restricted by OPMODE, and that because of this you should not recompile the ZPARM when migrating to MQ V9 LTS.

**More information**

There are a few things I haven’t covered here, such as the behaviour of OPMODE for queue managers in a queue sharing group (QSG), and also how OPMODE works for the MQ V9.0 continuous delivery releases. I hope to make those the subjects of future blog posts.

In the meantime, there’s more information on OPMODE in the Knowledge Center at the following locations:

* [OPMODE documentation](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.0.0/com.ibm.mq.mig.doc/q008200_.htm)
* [Changes to OPMODE in MQ V9.0.0](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.0.0/com.ibm.mq.pro.doc/q127165_.htm)

Thanks to Colin Paice and Ulrike Geuther who helped with this blog post.