

IBM MQ 7.x, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4 compatibility with previous versions
- including usage of CCDT files, JMS .bindings, SSL/TLS

<https://www.ibm.com/support/pages/node/100971>

Date last updated: 20-Jun-2024

Angel Rivera
IBM MQ Support

<https://www.ibm.com/products/mq/support>
Find all the support you need for IBM MQ

+++ Questions

Are MQ 7.0, 7.1, 7.5, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4 compatible with previous versions?
Can a recent MQ 9.4 client (for example) use a CCDT created with a previous version?
Can an older MQ 9.3 (for example) client use a CCDT created with a newer version?
Can a JMS .bindings file created in 9.3 be used with 9.4 (for example)?
Can a JMS .bindings file created in 9.4 be using with 9.3 (for example)?
Can an older MQ client using an old SSL protocol still work with MQ 9.x?

+++ Compatibility section in the online documentation for MQ

<https://www.ibm.com/docs/en/ibm-mq/9.4?topic=interoperability-application-compatibility-earlier-versions-mq>

IBM MQ / 9.4 / Maintaining and migrating / Migrating IBM MQ / Coexistence, compatibility, and interoperability /
Application compatibility and interoperability with earlier versions of IBM MQ

+ begin excerpt

Connecting an application that is built against libraries shipped with a later version of IBM MQ to an earlier version IBM MQ is not supported. Avoid building applications against a later version, and redeploying them to a queue manager running at an earlier version, although some applications do work in practice.

IBM MQ applications do interoperate with applications running on earlier versions of IBM MQ, as long as they use no new function. IBM MQ clients can connect to queue managers running at an earlier version than the client, as long as the client uses no new functions.

An IBM MQ application that uses only functions provided by an earlier version of a queue manager can continue to send messages to the earlier version. It does not matter what version of IBM MQ an application is built on and connected to. It can exchange messages with an application connected to an earlier version of IBM MQ, as long as it does not use new function.

Consider these four cases; the first two cases are not supported though they might work in practice, the last two cases are supported.

The first two cases require compatibility with an earlier version of IBM MQ.

The last two cases rely on the interoperability between all versions of IBM MQ

- Running an IBM MQ server application, built with a later version of IBM MQ, connecting to a queue manager running on a server with an earlier version of IBM MQ installed.
- Running an IBM MQ client application, built with a later version of IBM MQ, on a client platform with an earlier client installation, connecting to a queue manager running on a server with a later version of IBM MQ installed.
- Running an IBM MQ client application, built with a later version of IBM MQ, on a client platform with the later client installation, connecting to a queue manager running on a server with an earlier version of IBM MQ installed.
- Exchanging messages between an IBM MQ client or server application, connected to a queue manager running on a server with a later version of IBM MQ installed, with applications connected to a queue manager running on a server with an earlier version of IBM MQ installed.

Plan to avoid the first two cases, as they are not guaranteed to work all the time. If you are running an incompatible configuration and you encounter a problem, you must rebuild your applications with the correct level of IBM MQ. You can then continue with problem diagnosis.

+ end excerpt

+++ Summary

MQ 7.x, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4 queue managers and clients interoperate with queue managers and clients from any previous level of the WebSphere MQ or IBM MQ products (both LTS and CD).

This means that ...

- An MQ client at 7.0, 7.1, 7.5, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4 can connect to all queue managers, non-version 7, as well as version 7.0, 7.1, 7.5, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4.
- An MQ queue manager at 7.0, 7.1, 7.5, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4 can interact with all clients, non-version 7, as well as version 7.0, 7.1, 7.5, 8.0, 9.0, 9.1, 9.2, 9.3 and 9.4.
- The newer version of the code (queue manager / client) knows how to interact with previous versions and it will pretend to be at the same version of the other party. While the older party thinks that it is interacting with a component at the same version.
- If your 7.x, 8.0, 9.0, 9.1, 9.2, 9.3 or 9.4 client connects to an MQ 6.0 or earlier queue manager, or use a server-connection channel with the attribute SHARECNV = 0 (which treats the connection as being MQ 6.0), then you cannot take advantage of V7+ features and structures in your client application.
- See the CAVEATS in the next section.

+++ CAVEATS

- The MQ Development Team tries hard to maintain the compatibility between MQ versions; that is, the team is not purposefully trying to break the compatibility.
- Keep in mind that the older clients CANNOT exploit new features offered in newer MQ versions when dealing with a queue manager at a newer version.
- The testing of new version.releases is done against SUPPORTED versions, for example, the testing of MQ 9.4 did NOT include the testing with clients and queue managers at MQ 9.1 or earlier.
- It might be possible that overtime, the product and features have evolved and new changes may inadvertently break the compatibility with non-supported version.releases.
- If you are using unsupported versions of MQ, then we recommend that you do a thorough testing of your applications in a test environment between the old unsupported MQ components and the new and supported MQ components, just in case that a new feature has accidentally broken some aspects of the compatibility.
- It is a best practice to use the latest fix pack for each of the versions that you are using. Why? The latest fix packs could have fixed defects that are directly or indirectly related to compatibility.
- The compatibility mentioned in this document refers to:
Version.Release
... such as:
An MQ 9.3 client is compatible with an MQ 9.4 queue manager.
- The compatibility does not extend to a more granular specification:
Version.Release.Maintenance.FixPack (V.R.M.F) for Long Term Support (LTS)
Or
Version.Release.Maintenance (V.R.M) for Continuous Delivery (CD)
- If there are known problems at a given V.R.M.F (such as 9.3.0.1 LTS) then an update to a more recent fix pack for the same Version.Release will be necessary (such as 9.3.0.15 LTS). Notice that the update is within the same Version.Release (9.3) and a migration to a newer Version.Release is not necessarily required.

+++ Compatibility of MQ clients and CCDT files

Newer versions of the MQ clients (such as 9.4) know how to handle CCDT files that were created/edited by older queue managers (such as 6.0)

Older MQ clients (such as 6.0) do NOT know how to handle CCDT files that were created/edited by newer queue managers (such as 9.4). That is, older clients do not know what are the new attributes (if any) that were introduced in newer MQ versions.

As a result of APARs IT10863 and IT11547 which were added in MQ 7.5.0.7 and in 8.0.0.3, all supported versions of the MQ Clients are now capable of using CCDT files created by later product versions.

It should be noted that when a client uses a CCDT created with a later version of MQ, only channel attributes that exist at the client's earlier MQ version will be retrieved from the CCDT.

Similarly, if a client uses a CCDT created with an earlier version of MQ, only channel attributes that exist at the earlier MQ version will be present in the CCDT.

The following table takes into account the capability added in APARs IT10863 and IT11547 which are discussed later in this section.

MQ CCDT (created version)	Compatible MQ client versions (stated version is minimum level, includes later fix packs)									
6.0	6.0	7.0	7.1	7.5	8.0	9.0	9.1	9.2	9.3	9.4
7.0		7.0	7.1	7.5	8.0	9.0	9.1	9.2	9.3	9.4
7.1			7.1	7.5	8.0	9.0	9.1	9.2	9.3	9.4
7.5			7.1	7.5	8.0	9.0	9.1	9.2	9.3	9.4
8.0			7.1.0.8	7.5.0.7	8.0	9.0	9.1	9.2	9.3	9.4
9.0			7.1.0.8	7.5.0.7	8.0.0.3	9.0	9.1	9.2	9.3	9.4
9.1			7.1.0.8	7.5.0.7	8.0.0.3	9.0	9.1	9.2	9.3	9.4
9.2			7.1.0.8	7.5.0.7	8.0.0.3	9.0	9.1	9.2	9.3	9.4
9.3			7.1.0.8	7.5.0.7	8.0.0.3	9.0	9.1	9.2	9.3	9.4
9.3			7.1.0.8	7.5.0.7	8.0.0.3	9.0	9.1	9.2	9.3	9.4

Notes:

- There was no change in the MQCD length between 7.1 and 7.5. Therefore, the CCDT created on 7.5 can be used on 7.1.
- Unless documented otherwise, CCDTs created on higher level maintenance releases (for example 9.3.0.1) are compatible with lower level (for this example: 9.3.0.0) MQ clients running the same MQ version and release.

+ There are 2 APARs that allow OLDER MQ clients to use NEWER CCDT files!

These APARs relax the restriction by allowing newer CCDTs to be used on older clients but with the caveat that the older client cannot make use of any of the newer channel attributes. These attributes will assume their default values when the CCDT is negotiated with the queue manager.

For Java/JMS client applications:

<https://www.ibm.com/support/pages/apar/IT10863>

IT10863: WEBSHERE MQ CLASSES FOR JAVA/JMS APPLICATIONS CAN NOT USE CCDT FILES GENERATED ON A NEWER LEVEL QUEUE MANAGER

Version Maintenance Level

v7.0	7.0.1.14
v7.1	7.1.0.8
v7.5	7.5.0.6
v8.0	8.0.0.3

For C based (and non-Java/JMS) client applications:

<https://www.ibm.com/support/pages/apar/IT11547>

IT11547: WEBSHERE MQ CLIENT APPLICATIONS CANNOT USE CCDT FILES GENERATED ON A NEWER LEVEL QUEUE MANAGER.

This applies to C based and non-Java/JMS applications.

Version Maintenance Level

v7.1	7.1.0.8
v7.5	7.5.0.7
v8.0	8.0.0.3

+ Related references

a) MQ version 8 added support for using the runmqsc command in standalone mode to create a CCDT file without using a queue manager, using the "runmqsc -n" option:

<https://www.ibm.com/support/pages/node/616403>

New features added to the runmqsc command in IBM MQ Version 8.0

See:

Chapter 6: Flag -n to allow the modification of a local CCDT file

b) The default format for a CCDT is binary (non-human readable), but an alternative format which is human-readable text file was added in MQ 9.1.2 CD and it is available in MQ 9.2 LTS and CD and later.

<https://www.ibm.com/support/pages/node/6568807>

Using IBM MQ CCDT file in JSON format

+++ Compatibility of MQ clients and JMS .bindings files

- A JMS .bindings file that was created at an older version (such as 9.3) than the MQ JMS client (such as 9.4) can be used without version related errors by the client.

- A JMS .bindings file that was created at a newer version (such as 9.4) than the MQ JMS client (such as 9.3) can be used without version related errors by the client.

If there are any new attributes in the JMS administered objects that the older client cannot read, they are ignored.

+++ Compatibility of SSL/TLS cipher specifications

There have been changes in recent fix packs and version.releases related to the compatibility of SSL/TLS cipher specifications.

For example, if you have a MQ Client at 7.1.0.1 connecting to a MQ 8.0.0.1 queue manager using an older SSL cipher specification, after applying Fix Pack MQ 8.0.0.2 to the queue manager, the connectivity may be broken because the queue manager indicates that the older SSL cipher spec is now deprecated.

+ Deprecated CipherSpecs

<https://www.ibm.com/docs/en/ibm-mq/9.4?topic=cipherspecs-deprecated>

IBM MQ / 9.4

Deprecated CipherSpecs

+ For more details on the compatibility of SSL/TLS cipher specifications, see the following blog entry for MQ 7.x and MQ 8.0

<https://www.ibm.com/support/pages/ssl-and-tls-cipher-specification-deprecations-mq-product>

SSL and TLS Cipher Specification Deprecations for the MQ Product

Miguel A. Rodriguez, May 2016

Due to the recent security vulnerabilities (for example, POODLE Attack), the latest MQ product Fix Packs come with stricter, default security requirements that affect the use of the compromised Secure Socket Layer (SSL) and Weak Transport Level Security (TLS) Cipher Specifications. Since these Cipher Specification deprecations are disabled in MQ Fix Packs by default, review the article for the changes separated by MQ versions and Fix Pack levels.

+++ end