

WebSphere Application Server and IBM MQ do not agree on the number of JMS connections

<https://www.ibm.com/support/pages/node/81567>

Date last updated: 15-Feb-2023

IBM MQ Support

<https://www.ibm.com/products/mq/support>

Find all the support you need for IBM MQ

+++ Problem +++

The WebSphere Application Server JMS connections to IBM MQ does not equal the number of Server Connection channel instances in IBM MQ.

+++ Cause +++

This is working as designed.

+++ Resolving The Problem +++

When an MQ Client application is running in WAS using the MQ JMS Provider and is connected to an MQ queue manager, there is no direct correlation between the number of JMS Client connections in to the IBM MQ queue manager and the number of Server Connection channels running on the queue manager.

Why? Because this is a side effect of connection caching.

On the client side there is caching of connections that happens in both the client and WebSphere Application Server:

1. In the MQ Java™ Client (MQQueueManager pool)
2. WebSphere Application Server Connection Handles (WAS JCA Pool)

The number of Server Connection channels on the MQ queue manager will grow depending on the load.

When the connection is closed, the connection handle associated with that connection is released and sent back into the WAS JCA pool.

However, even after they are cleaned up from the JCA pool, it does not mean the number of active channels on the MQ queue manager will drop. This is because the objects that constitute the network presence are still cached at the Java client level (MQQueueManager objects).

The reap of these objects is performed by the MQSimpleConnectionManager object.
See:

<https://www.ibm.com/docs/en/ibm-mq/9.3?topic=applications-connection-pooling-in-mq-classes-java>

IBM MQ / 9.3

Connection pooling in IBM MQ classes for Java

This destroys connections on a least-recently-used basis. A connection is destroyed if it has not been used for 5 minutes, or if there are more than 10 unused connections in the pool.

Because of preemptive caching, the number of active channels will be a little more than the highest number of Connections used simultaneously in the last minutes:

(JCA reap time + MQ Java reap time)

In short, because of the way connections are cached and managed in both the MQ Client code and the WAS JCA layer the number of active connections on the Client side may not be the same as the number of Server Connection channels on the MQ queue manager side.

With regard to the attribute MaxChannels in the queue manager, there is no way to predict exactly what value to set this to at any given moment due to the dynamic nature of connections/disconnections and caching.

However, the default settings for a Queue Connection Factory (QCF) are:

Maximum Connection Pool Size = 10

Maximum Session Pool Size = 10

Each Client connection has its own session pool. Therefore, with the default settings above, each connection will get 10 sessions.

Since there are 10 connections in the pool and 10 sessions per connection this permits a maximum of 100 actual socket connections to MQ.

Therefore MaxChannels on the MQ Queue Manager should be AT LEAST this value (100).

A safe practice would be to set MaxChannels to a higher number other than the default 100, perhaps to 1000, although you may want to use a smaller value if your system is resource constrained.

If you change the Connection Pooling default values be sure to adjust MaxChannels accordingly.

For more information on setting MaxChannels, see:

<https://www.ibm.com/docs/en/ibm-mq/9.3?topic=information-channels-stanza-qmini-file>

IBM MQ / 9.3

Channels stanza of the qm.ini file

+++ end +++