

Collection of FAQs and articles regarding IBM MQ Channels and Connections

<https://www.ibm.com/support/pages/node/6984195>

Date last updated: 6-Feb-2024

Angel Rivera
IBM MQ Support

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

Find all the support you need for IBM MQ

+++ Objective +++

This document provides a compilation of Frequently Asked Questions (1st section of this document) and articles (2nd section of the document) regarding IBM MQ Channels and Connections.

+++ SECTION 1: Frequently Asked Questions

++ Very common question about stopping connections

<https://www.ibm.com/support/pages/node/616249>

How to identify IBM MQ client connections and stop them

++ The following is useful for beginners:

<https://www.ibm.com/support/pages/node/7029698>

Common errors during initial connectivity between an IBM MQ client using MQSERVER and a remote queue manager

++ Important distinction between a "connection" and a "channel instance" (MaxChannels)

The following article provides a good explanation of a common misunderstanding:
<https://mqgem.wordpress.com/2016/06/23/maxchannels-vs-dis-qmstatus-conns/>
MaxChannels vs DIS QMSTATUS CONNS
(Article from Morag Hughson)

Based on that article, here are some important clarifications:

a) Total Connections: local (non channel) and client (using a channel instance)

```
DISPLAY QMSTATUS CONNS
```

This runmqsc command shows how many connections are currently made into the queue manager.

This is a count of the number of applications (or some queue manager processes too) that have made an MQCONN/MQCONN to the queue manager.

These connections might be local or client connections (which use a channel instance) - both types contribute to the total.

To see the local connections (no channel) use:

```
DISPLAY CONN(*) ALL WHERE (CHANNEL EQ ' ')
```

To see the remote connections (using channel) use:

```
DISPLAY CONN(*) ALL WHERE (CHANNEL NE ' ')
```

The above commands might be confusing because they use "negative logic".

For example, if the attribute CHANNEL is null/empty (represented by ' ' in the WHERE clause), then there is NO channel being used, and it is a local connection.

On the other hand, if the attribute CHANNEL is NOT NULL, that is, there is a string in the attribute CHANNEL, then there is a channel being used.

b) Client Connections that use a channel instance and contribute to MaxChannels

```
DISPLAY CHSTATUS(*) CURSHCNV
```

This runmqsc command shows how many running channel instances are made to the queue manager.

If you add up all the numbers shown in CURSHCNV, then this total will be less than (or equal to) the number of channel-based connections that are listed when:

```
DISPLAY CONN(*) ALL WHERE (CHANNEL NE ' ')
```

Both queue manager channels and client channels contribute to that total.

This is the number to compare against the attribute MaxChannels of the qm.ini for the queue manager (the default is 100 and it is not explicitly included in the qm.ini).

++ I just created a new queue manager: why does it show Active connections even though I do not have MQ client applications running?

You created a new queue manager.
You are using the MQ Explorer to monitor the queue manager.
You have not started MQ client applications.
You see that there are active connections.

Why?

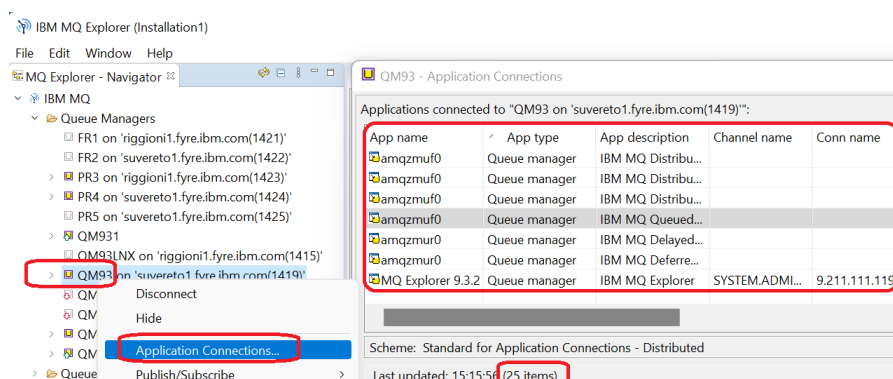
a) The MQ Explorer is considered to be an MQ Client application, therefore, it is expected to see an active connection for it.

b) The "queue manager" is really a collection of interconnected processes and programs that communicate using Inter-Process-Communication (IPC) such as shared memory and semaphores. That is, the "queue manager" is not a monolithic program. These processes and programs that constitute the queue manager, are shown as active connections.

c) The display of the connections includes all types of connections, applications and processes:

- processes for the MQ queue manager proper (the queue manager is a collection of several processes)
- local applications that use bindings / shared memory,
- MQ client applications that use a server-connection channel
- sender/receiver channels to communicate with other queue managers

For example, a newly created queue manager with no activity, and the MQ Explorer connected to it, is already consuming:
25 connections



By the way, the equivalent runmqsc command is:
DISPLAY CONN(*)

+ Related question:

If there are 25 connections listed under "Application Connections..." and DISPLAY CONN(*), and if the default Maximum Channels is 100, does it mean that only 75 new connections can be made?

Notice that this question is not a proper one because it is mixing two different concepts: channels and connections.

The attribute Maximum Channels (MaxChannels) limits the number of channel instances, but does not limit the number of connections.

There is no attribute in the queue manager that limits the number of connections.

A local bindings application when connected to the queue manager, is establishing a connection.

But this local bindings application is NOT consuming a "channel" and thus, it is not affecting the count for Maximum Channels (MaxChannels).

For the question on this section, if there is only 1 channel instance running (the one for the MQ Explorer), then only 1 channel instance out of the MaxChannels of 100 is being used, and you could have up to 99 other new channel instances.

++ Error AMQ9513E and how to increase MaxChannels

On Distributed platforms the default value for MaxChannels is 100.

A system that is busy serving connections from the network might need a higher number than the default setting, otherwise you will see the following entry in the error log of the queue manager:

```
AMQ9513E Maximum number of channels reached.
```

Determine the value that is correct for your environment, ideally by observing the behavior of your system during testing.

On Distributed platforms, MaxChannels is an attribute in the qm.ini file.

Example for changing maximum channels for a queue manager from the default 100 to 300:

Step 1: Modify the qm.ini of the queue manager:

```
CHANNELS:
```

```
MaxChannels=300
```

Step 2: Stop and restart the queue manager for the change in the qm.ini to take effect.

See:

<https://www.ibm.com/support/pages/node/408395>

Finding the number of running channels in an IBM MQ queue manager

Tags: AMQ9513E, MaxChannels

<https://www.ibm.com/support/pages/node/236995>

IBM MQ error AMQ9513E, Maximum number of channels reached

++ Finding time stamp for a Connection

<https://www.ibm.com/support/pages/node/6207043>

How to find the date-time stamp when an MQ connection started (runmqsc DISPLAY CONN, DISPLAY CHSTATUS, plus amqldmpa)

The objective of this tutorial is to provide you with 2 procedures to identify the oldest connection.

- a) The ConnectionID is based on the start time of the connection.
- b) Using runmqsc and amqldmpa

++ Common question regarding the command START CHANNEL

<https://www.ibm.com/support/pages/node/241881>

Why is the channel status not RUNNING after issuing START CHANNEL command on an IBM MQ receiver or server-connection channel

In short:

The START CHANNEL command only enables the channel.

These channels will be in status RUNNING only when these channels are actually being used by an active connection, that is, there is an MQ Client application currently connected and using the channel.

If you start the channel but no application is using it, then the DISPLAY CHSTATUS will indicate that a running instance for this channel is not found.

++ Displaying MaxChannels at runtime, from an FDC, such as one generated by amqrmppa

Scenario: It might be possible for one MQ admin to modify the qm.ini to increase the MaxChannels, but forgets to restart the queue manager.

For example, the initial value was 100 and the update in the qm.ini was 200.

This means that the running queue manager is still using the old value of 100 and not using the new value in the qm.ini of 200.

There might be a confusion because the actual behavior (no new connections can be made after reaching 100). Another MQ admin may say: hum, the qm.ini says 200, then why are we getting the error about MaxChannels has been reached when we have only 100 connections?

The question is: is there a way to find out the real MaxChannels value at the time that queue manager started?

<https://www.ibm.com/support/pages/node/872924>

How to find out the value of MaxChannels that the queue manager is actually using at runtime (from an FDC)

Search for the section:

Channel Status Table Header

See the value for MaxChannels that the queue manager is actually using (in this test is 100):

Channel Status Table Header

```
{
  Version                4
  HeaderSize             456
  ChanSeqNo              123
  AllocatedCount         1
  MaxChannels            100    <==*** This is the entry that shows
                               the original value for MaxChannels
  MaxActiveChannels      100
```


++ Displaying how many channels are in use

The following excerpt is from:

<https://www.ibm.com/support/pages/node/408395>

Finding the number of running channels

To display all the channels, use:

Windows: you can enter the following compound commands.

Ensure to replace QMGR with the proper values for your system:

```
echo DISPLAY CHSTATUS(*) | runmqsc QMGR | find /c "AMQ8417"
```

UNIX: you can enter the following compound commands.

Ensure to replace QMGR with the proper values for your system:

```
echo "DISPLAY CHSTATUS(*)" | runmqsc QMGR | grep -c 'AMQ8417'
```

Note: Thanks to David Awerbuch for pointing out this option of grep.

If the version of grep in your system does not support the flag for counting (-c) you can use:

```
echo "DISPLAY CHSTATUS(*)" | runmqsc QMGR | grep 'AMQ8417' | wc -l
```

For UNIX:

One possible way to apply this information is to create a cron job that issues the mentioned command and if the number of running channels is getting closer to the MaxChannels, then an MQ Administrator might be notified. The following technote could be used as a reference for creating a cron job:

<https://www.ibm.com/support/pages/node/87035>

cron job for clearing all messages from a queue

++ Identify the application that has opened a queue and is still using it

<https://www.ibm.com/support/pages/node/358665>

Identify the application that has an IBM MQ queue opened

The following example is for queue "Q1". The APPLTAG and CONNAME provide the name of the application and the IP address.

```
display conn(*) where(objname eq Q1) type(*) all
```

Example output:

```
AMQ8276I: Display Connection details.
CONN(D081D56201CF3624)
EXTCONN(414D5143514D44454D4F2020202020)
TYPE(*)
PID(77538)                                TID(14)
APPLDESC(IBM MQ Channel)                 APPLTAG(amqsputc)
APPLTYPE(USER)                           ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN)              CLIENTID( )
CONNAME(9.46.66.142)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(mqm)                               UOWLOG( )
UOWSTDA( )                                UOWSTTI( )
UOWLOGDA( )                               UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0)                               UOWSTATE(NONE)
CONNTAG(MQCTD081D56201CF3624QMDEMO_2022-07-18_08.52.23amqsputc)

OBJNAME(Q1)                               OBJTYPE(Queue)
ASTATE(NONE)                              HSTATE(INACTIVE)
OPENOPTS(MQOO_OUTPUT,MQOO_FAIL_IF QUIESCING)
READA(NO)
```

The following fields are of interest:

PID(77538) => Process id (if application is local)

APPLTAG(amqsputc) => name of the application

CHANNEL(SYSTEM.DEF.SVRCONN) => server-connection channel

CONNAME(9.46.66.142) => IP address (aka, connection name)

USERID(mqm) => Userid that has invoked the application

++ Best practice of creating a dedicated server-connection channel for applications

The default server-connection channel SYSTEM.DEF.SVRCONN is one that is intended to be used for sporadic connections, such as a testing of samples or for activities that are intermittent or of short duration.

But if a client application is going to be connected constantly to the queue manager, then it is best to create and use a dedicated server-connection channel for that application.

There are some benefits for doing so:

- If the channel needs to be stopped, then only the dedicated application will be affected.
- You can exploit a customized "channel authentication record", which is a feature introduced in MQ 7.1.
- You could have different characteristics for the channel, such as SSL, MAXMSGL (maximum message length), MAXINST (maximum instances) and MAXINSTC (maximum instances per client).

- To identify the amqrmppa process servicing a server-connection channel:

<https://community.ibm.com/community/user/integration/blogs/morag-hughson/2021/09/28/ibm-mq-little-gem-53-jobname-in-channel-status>

IBM MQ Little Gem #54: JOBNAME in Channel Status

(By Morag Hughson)

The JOBNAME field of the channel status entry for each SVRCONN channel instance (TCP connection) describes the process ID (and thread) of the process servicing that connection

- The following tutorial is very comprehensive, and includes workarounds for Test queue managers regarding channel authentication records.

It provides the steps to allow a non-Administrator user (fulano) to use a server-connection channel and to put/get messages from a queue.

The authorizations needed for MQ Explorer are also included.

Even though the focus is for Linux, the actual MQ steps are the same for Windows.

<https://www.ibm.com/support/pages/node/1135522>

Configuring an IBM MQ queue manager to use a dedicated Listener, Channel and Queue in Linux

+++ SECTION 2: Related articles

<https://www.ibm.com/support/pages/node/7053159>

Can an IBM MQ Channel have multiple connection instances?

Answer: Yes.

This article provides clarifications and details while using a server-connection channel "MY.CHANNEL" from different hosts, using the sample "amqspc" which uses a network connection.

<https://www.ibm.com/support/pages/node/152555>

MQ Commands to setup two way communication between two queue managers on Sender and Receiver channels

- It has a section to enable Triggering of the Sender Channels to start when a message arrives

(Updated with figure and references)

Two files are provided that can be used to generate the runmqsc commands mentioned in this document:

1 sample Windows batch file: gen-mqsc-2-qmgrs.bat

1 sample Unix shell script: gen-mqsc-2-qmgrs.sh

tag: two-ways

<https://www.ibm.com/support/pages/node/7004685>

IBM MQ error AMQ9209E 'Connection to host X for channel Y closed', technical details and how to manually generate it in Linux and AIX

<https://www.ibm.com/support/pages/node/341913>

IBM MQ AMQ9526E Message sequence number error for channel

<https://www.ibm.com/support/pages/node/6985635>

How to specify the ConnectionNameList for an IBM MQ multi-instance queue manager in MQSERVER, CCDT, Explorer, JMS ConnectionFactory

Note: The same principle applies for the CONNAME field for a Sender channel that connects to a multi-instance queue manager.

<https://www.ibm.com/support/pages/node/664115>

AMQ9206 or AMQ9208 ECONNRESET from TCP/IP, Linux 104 (x68), Windows 10054 (x2746), on an IBM MQ client channel connection to a local server
(Tags: AMQ9206I or AMQ9208I)

<https://www.ibm.com/support/pages/node/6983857>

Example of attributes MAXINST and MAXINSTC to better allocate the active instances for your server-connection channels in IBM MQ

<https://www.ibm.com/support/pages/node/400151>

Instructions and Warnings: How to increase the MAXMSGL for IBM MQ Queue Managers, Queues and Channels from the default 4 MB to a higher number

<https://www.ibm.com/support/pages/node/6570847>

MQ errors AMQ9219E and AMQ9213, due to TCP error ENOTCONN (Linux 107, AIX 76, Windows 10057)

<https://www.ibm.com/support/pages/node/514147>

Example of MQ sender channel attribute LOCLADDR to determine the range of ports used for outbound TCP communications

<https://www.ibm.com/support/pages/node/6415877>

If the MQ SYSTEM.*.SVRCONN channels are not used, is it OK to specify MCAUSER('NOBODY')?

<https://www.ibm.com/support/pages/node/874402>

Should the server-connection (SVRCONN) channels be disconnected when the MQ Listener is down?

<https://www.ibm.com/support/pages/node/241885>

Sender-Receiver behavior when disconnect interval stops the channel pair: AMQ9002, AMQ9545 and AMQ9001

<https://www.ibm.com/support/pages/node/243831>

What is the order for terminating the active channels when a queue manager is ending?

<https://www.ibm.com/support/pages/node/213249>

How to identify the IP address of a client connected via a server connection channel

<https://www.ibm.com/support/pages/node/7026591>

Comparison in behavior of IBM MQ samples amqsputc and amqsphac and variants during automatic client reconnection with a multi-instance queue manager

<https://www.ibm.com/support/pages/node/7025442>

Sample Linux script to periodically capture the connections to an IBM MQ queue (mq-capture-conn-for-queue)

<https://www.ibm.com/support/pages/node/6985635>

How to specify the ConnectionNameList for an IBM MQ multi-instance queue manager in MQSERVER, CCDT, Explorer, JMS ConnectionFactory

Note: The same principle applies for the CONNAME field for a Sender channel that connects to a multi-instance queue manager.

<https://www.ibm.com/support/pages/node/6561647>

AMQ9271E error messages for MQ client connections: [TIMEOUT] 65 seconds
tag: hbint, heartbeat, disconnect

<https://www.ibm.com/support/pages/node/6430633>

Connection precedence hierarchy for MQ Clients: PreConnect Exit, MQCONN API, MQSERVER, mqclient.ini, CCDT

<https://www.ibm.com/support/pages/node/196563>

MQ RC 2035 MQRC_NOT_AUTHORIZED or AMQ4036 or JMSWMQ2013 when using client connection as an MQ Administrator (CHLAUTH, channel authentication records)

<https://www.ibm.com/support/pages/node/6207335>

How to find out the process id of the MQ amqrmppa process that is handling a specific client connection

+++ end +++