

IBM MQ error AMQ9513E, Maximum number of channels reached

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

Date last updated: 27-Apr-2023

Angel Rivera
IBM MQ Support

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

Find all the support you need for IBM MQ

+++ Question +++

You would like to know more details regarding the following MQ error message in the error log of the queue manager:

AMQ9513E: Maximum number of channels reached

While the MQ Client application gets reason code:

2537 MQRC_CHANNEL_NOT_AVAILABLE

+++ Answer +++

The default value for MaxChannels in the queue manager configuration file (qm.ini) is: 100

The value for this attribute is only for connections that use channel instances.

That is, those connections that use local bindings and which are not using channels, are not included in the evaluation of MaxChannels.

The error AMQ9513E (Maximum number of channels reached) is reported in the MQ queue manager error logs when a client application attempts to connect to the queue manager via the network using a channel, but the maximum number of channels are in use.

The result is that the new attempt to connect (that is, to start a new instance of the channel) will fail.

This failure does not affect the status of instances of channels that are currently running.

There are three causes for this error message.

For items 2 and 3 below, the MQ administrator needs to contact the application administrators or developers to determine the maximum number of channels needed and if the application needs to be changed.

1) The queue manager attribute `MaxChannels` is set too low for the required number of client connections (default is 100).

This attribute is specified in the Channels stanza in the `qm.ini` file of the queue manager.

If you determine that the value set for `MaxChannels` is not sufficient, then increase the value and restart the queue manager, for the new value to take effect. This will reserve the needed system resources. This may also provide a work-around to an application problem until it can be fixed.

In Unix the `qm.ini` is located at:

```
/var/mqm/qmgrs/<QMGRNAME>/qm.ini
```

In Windows:

```
C:\Program Files\IBM\MQ\qmgrs\<QMNAME>\qm.ini
```

Example of setting `MaxChannels` set to 300 in the Channels stanza inside the `qm.ini` file:

Channels:

```
MaxChannels=300
```

2) Client applications are holding the channels longer than needed, thereby not freeing up the channels for other application to use.

The following article could be helpful in case that there are applications that might not behaving correctly and spawning many instances that might affect `MaxChannels`.

<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

3) Client applications are not freeing the channels via `MQDISC` before terminating. This can occur if the application, or client system, is killed or if there are network glitches.

In MQ 6.0, this item was a leading cause for reaching the max channels condition.

"Orphaned" channels were not killed by the queue manager because the queue manager could not communicate with the client due to the half-duplex nature of server-connection channels in V6.

In MQ 7.x and later, with the introduction of full-duplex channels, the queue manager can now ask periodically via a heartbeat if the client at the other end of the channel is still alive. If not alive anymore, then the connection is terminated, eliminating "orphan" channels.

However, even when using MQ 7.x or later, if the server-connection channel has SHARECNV equal to 0, then the channel is effectively working in half-duplex mode, and it could cause "orphaned" channels.

The same happens if the client specifies PROVIDERVERSION equals 6 (for version 6).

If using SHARECNV = 0 (or PROVIDERVERSION 6), then the MQ administrator can implement KeepAlive. This will bring down any channels where the originating application/system has terminated or where a network problem terminated the connection.

The ClientIdle parameter is recommended for half-duplex channels (when SHARECNV=0). When using full-duplex channels (SHARECNV of 1 or larger), there is no need for it. It is an undocumented attribute for the qm.ini file.

+ Typical scenario

An MQ Client application tries to connect to the queue manager, but fails with rc 2537

```
mqm@bilbao1.fyre.ibm.com: /home/mqm  
$ amqsputc Q1 QM93  
Sample AMQSPUT0 start  
MQCONN ended with reason code 2537
```

To find out the name associated with this reason code, you can use:

```
$ mqrc 2537  
2537 0x000009e9 MQRC_CHANNEL_NOT_AVAILABLE
```

Then the MQ Administrator looks at the bottom of the error log for the queue manager and sees the following:

AMQ9513E: Maximum number of channels reached.

EXPLANATION:

The maximum number of channels that can be in use simultaneously has been reached. The current defined maximum number of channels is '5'. The number of permitted channels is a configurable parameter in the queue manager configuration file.

ACTION:

Wait for some of the operating channels to close. Retry the operation when some channels are available.

+ Recommended articles

The following tutorial has a lot of information regarding channels and connections:

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

How to identify MQ client connections and stop them

https://www.mqtechconference.com/sessions_v2014/KeepingChls.pdf

Keeping MQ Channels Up and Running

Paul Clarke

+ Is there a performance impact in using a high value for MaxChannels?

Increasing MaxChannels/MaxActiveChannels will only require additional resources 'on demand' when the additional connections are made to the queue manager.

In other words, a queue manager started with a MaxChannels value of 600 but only 100 channels are actually connecting, will use the same amount of memory as having MaxChannels value of 100 and only 100 channels are actually connecting. That is, the effective resources that are used are only for 100 channels.

As another example: if MaxChannels is increased from 600 to 950, then a queue manager started with a MaxChannels value of 950 and connecting 600 channels will use the same amount of memory as connecting 600 channels to a queue manager with a MaxChannels value of 600.

HOWEVER ...

While an additional channel status entry will be allocated for each new channel status entry, there will also be an increase in the number of queue manager and MCA threads. Each new connection requiring additional 188 KB of memory. Additional CPU and memory would be required to service the additional workload.

<https://www.ibm.com/support/pages/node/582345>

The MQV8 performance report for AIX:

This report contains a section on memory usage (section 3.6) indicating that additional client connections in the benchmarking tests required approximately additional 188 KB each.

+ Common mistake of having 2 "Channels" stanza

A common typo mistake when modifying the qm.ini is to have 2 "Channels" stanzas. Usually, the MQ administrator when reviewing this file, "sees" only the 1st entry in the top portion of the file, and fails to see the 2nd entry at the bottom. Keep in mind that the 2nd entry will be the "winner" and the first entry will be ignored.

The following is an example of a wrongly configured qm.ini file, with 2 "Channels" stanzas.

The first one has MaxChannels=1000 (which is the desired value) and the second has MaxChannels=200.

During the startup, the queue manager uses the default value of 100 for MaxChannels, and if the qm.ini has defined this attribute, then the queue manager will read the 1st entry, setting internally MaxChannels=1000 and overwriting the default of 100, but then the queue manager reads the 2nd entry, setting internally the value 200 and overwriting the previous value of 1000, with the result that the real value that will be used at runtime will be 200. The end result is that the 1st entry (1000) is ignored.

Channels:

MaxChannels=1000

Service:

Name=AuthorizationService

EntryPoints=14

Channels:

MaxChannels=200

+ Finding the number of running channels

MQ does not have utilities that will alert the MQ administrator in case that the maximum channels limit is being reached or has been reached.

One possibility is for someone in your team to create a cron job that issues the commands mentioned in the following technote and then if the number of running channels is getting closer to the MaxChannels, then an MQ Administrator might be notified.

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

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

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 in an MQ queue manager

+ Scenario: qm.ini is modified but queue manager is not restarted

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)

From an FDC, such as one generated by amqrmppa

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
  AllocatedCount         1
  MaxChannels            100 <== This is the entry that shows the MaxChannels !!
  MaxActiveChannels     100
```

+++ end +++