IPPROCS or OPPROCS greater than 0 in an IBM MQ queue prevents normal termination of queue manager by "endmqm QmgrName" (Need to use: endmqm -i QmgrName)

https://www.ibm.com/support/pages/node/163087

Date last updated: 18-Apr-2023

Angel Rivera
IBM MQ Support
https://www.ibm.com/products/mq/support
Find all the support you need for IBM MQ

+++ Problem +++

You notice that when the IBM MQ attributes IPPROCS or OPPROCS for a queue are greater than 0, and then you try to end the queue manager by issuing:
  endmqm QmgrName
… then the queue manager goes into "quiescing" mode but it does not fully terminate.
Is there a way to speed up the shutdown of the queue manager?

+ Resolving The Problem +

The recommended way to end the queue manager and avoid a potential long quiescing state is to specify the flag -i (immediate):

- For Standalone queue managers,
  endmqm -i QmgrName

- For Multi-Instance queue managers:
  endmqm -is QmgrName

+++ Background +++

++ Queue attributes IPPROCS and OPPROCS

Summary for these queue attributes:
IPPROCS = MQ GETs
OPPROCS = MQ PUTs

One potential item that could cause confusion is the terminology for IPPROCS (Input, MQGETs) and OPPROCS (Output, MQPUTs).

A critical point to keep in mind is that the designation of "what is input" and "what is output" is from the perspective of the MQ CLIENT APPLICATION (that is, NOT from the perspective of the queue manager).

The attribute IPPROCS shows the count of those MQ Client Applications that have opened the Queue for INPUT, that is, the application is going to do an MQ GET.

Similarly, OPPROCS shows the count for MQ Client Applications that have opened the Queue for OUTPUT, that is, for doing an MQ PUT.

++ Command endmqm

If "endmqm QMgrName" does not effectively end the queue manager, you can try to speed up the shutdown process by adding the flag -i such as:
endmqm -i QMgrName

The default (endmqm QMgrName) is to use "-c" as in:
endmqm -c QMgrName

-c Controlled (or quiesced) shutdown:

This is the default.

This is the reply from the command:

```
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
```

The queue manager stops, but only after all applications have disconnected.
Any MQI calls currently being processed are completed.

Note: If an application has opened a queue for put or for get, but still is connected, then this may prevent the queue manager from ending. It will remain in "quiescing".

This default behavior can be "too polite" and may prevent to end the queue manager when you really want to end it.
If you have an MQ JMS client running in an application server and which is connected to the

queue manager via a server-connection channel, then as long as this MQ JMS client is connected, you cannot terminate the queue manager via the default flag. This situation includes when there are NO messages to be retrieved by the MQ JMS client.

Thus, in this case your queue manager is prevented from terminated by an application that is not really active!

For Multi-Instance queue managers, if you use a Controlled flag to terminate an Active instance in order to do a switchover, the result is that the queue manager has a status of Quiescing but does NOT terminate or that takes a very long time to terminate.
Why? Because the default behavior is to wait for the MQ client application that is connected to disconnect by itself.

And because the Active instance has NOT fully ended, the Standby still remains as Standby and there is no switch over.
In case that after a long time the Active instance finally terminates, then the Standby will be able to become the new Active instance.

-i Immediate shutdown:

The queue manager stops after it has completed all the MQI calls currently being processed.
Any new MQI requests issued after the endmqm command has been issued, will fail.
Any incomplete units of work are rolled back when the queue manager is next started.
Control is returned after the queue manager has ended.
.
Note: If a client application has opened a queue and still is connected, such as the MQ Explorer, then this option will terminate the server-client connection, allowing the queue manager to terminate.
.
This flag -i is "polite", but it is more assertive (but not aggressive)
It does NOT terminate the queue manager in the middle of an active MQI call (this drastic action is only done when using the -p flag: Pre-emptive shutdown).
With the -i flag the queue manager waits politely for the activity to finish, and then, if the MQ client is no longer "active", but just connected, then the queue manager will gracefully disconnect the MQ client.
.
Then the recommended way to end the queue manager is:
- For Standalone queue managers,
  endmqm -i QmgrName
- For Multi-Instance queue managers:
  endmqm -is QmgrName

.
In short: because the default endmqm behavior is too polite and may prevent the queue manager from actually terminating, the flag -i (immediate) allows an assertive but graceful disconnection of inactive MQ clients, allowing the queue manager to end timely.

-p Preemptive shutdown:

WARNING: !!!! Use this type of shutdown only in exceptional circumstances. For example, when a queue manager does not stop as a result of a normal endmqm command. !!!

The queue manager might stop without waiting for applications to disconnect or for MQI calls to complete.
This can give unpredictable results for IBM MQ applications.
The shutdown mode is set to immediate shutdown. If the queue manager has not stopped after a few seconds, the shutdown mode is escalated, and all remaining queue manager processes are stopped.

+++ Scenario that shows quiescing mode from "endmqm" without the -i flag

Queue manager QM93LNX is running IBM MQ 9.3 in RHEL, in host "riggioni1", port 1415.

A queue Q1 has not been opened by any application.
Thus, IPPROCS and OPPROCS are 0:

In one command terminal window enter:

```
mqm@riggioni1.fyre.ibm.com: /home/mqm
$ strmqm QM93LNX
$ runmqsc QM93LNX
DISPLAY QSTATUS(Q1) IPPROCS OPPROCS
AMQ8450I: Display queue status details.
   QUEUE(Q1)                               TYPE(QUEUE)
   CURDEPTH(0)                             IPPROCS(0)
   OPPROCS(0)
```

In another command terminal window, the MQ sample command "amqsput" is issued to put a message. This command uses local bindings.

```
mqm@riggioni1.fyre.ibm.com: /home/mqm
$ amqsput Q1 QM93LNX
Sample AMQSPUT0 start
target queue is Q1
```

(DO NOT EXIT! This sample needs to be connected!)

Notice that the text of the message has not being entered yet.
The application is just waiting for the user to enter the text.
The application has already opened the queue and thus, the counter for OPPROCS has been incremented from 0 to 1:

```
$ runmqsc QM93LNX
DISPLAY QSTATUS(Q1) IPPROCS OPPROCS
AMQ8450I: Display queue status details.
   QUEUE(Q1)                               TYPE(QUEUE)
   CURDEPTH(0)                             IPPROCS(0)
   OPPROCS(1)
```

In another window from another host, the MQ sample command "amqsgetc" is issued to get a message:

```
mqm@fosdinovo1.fyre.ibm.com: /home/mqm
$ export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/riggioni1(1415)'
```

```
$ amqsgetc Q1 QM93LNX
Sample AMQSGET0 start
```

Notice that now the counter for IPPROCS has been incremented from 0 to 1, and that OPPROCS is still at 1 because amqsput is still connected:

```
$ runmqsc QM93LNX
DISPLAY QSTATUS(Q1) IPPROCS OPPROCS
AMQ8450I: Display queue status details.
   QUEUE(Q1)                               TYPE(QUEUE)
   CURDEPTH(0)                             IPPROCS(1)
   OPPROCS(1)
```

But 30 seconds later, amqsgetc terminates because no messages arrived within 30 seconds:

```
mqm@fosdinovo1.fyre.ibm.com: /home/mqm
$ amqsgetc Q1 QM93LNX
Sample AMQSGET0 start
no more messages
Sample AMQSGET0 end
```

Now the IPPROCS counter decreased from 1 to 0.
.
But amqsput is still connected, waiting for the user to enter a message and thus OPPROCS is still 1.

```
$ runmqsc QM93LNX
DISPLAY QSTATUS(Q1) IPPROCS OPPROCS
AMQ8450I: Display queue status details.
   QUEUE(Q1)                               TYPE(QUEUE)
   CURDEPTH(0)                             IPPROCS(0)
   OPPROCS(1)
```

Let's assume that we wanted to terminate the queue manager, keeping in mind that amqsput is still running and connected to the queue (OPPROCS is still 1):

```
mqm@riggioni1.fyre.ibm.com: /home/mqm
$ endmqm QM93LNX
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
```

Notice the status of the queue manager:

```
mqm@riggioni1.fyre.ibm.com: /home/mqm
$ dspmq -m QM93LNX
QMNAME(QM93LNX)                                   STATUS(Quiescing)
```

Notice that we cannot make new connections, not even running runmqsc:

```
mqm@fosdinovo1.fyre.ibm.com: /home/mqm
$ amqsgetc Q1 QM93LNX
Sample AMQSGET0 start
MQCONNX ended with reason code 2538

mqm@fosdinovo1.fyre.ibm.com: /home/mqm
$ mqrc 2538
     2538  0x000009ea  MQRC_HOST_NOT_AVAILABLE
```

Note: Depending on the version.release, you may get the reason code 2161.

```
$ mqrc 2161
     2161  0x00000871  MQRC_Q_MGR_QUIESCING
```

The monitoring of the MQ processes for this queue manager does not show any changes and the processes are still running.

```
mqm@riggioni1.fyre.ibm.com: /home/mqm
$ ps -ef | grep QM93LNX
mqm       2634     1  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzxma0 -m QM93LNX -u
mqm
mqm       2644   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzfuma -m QM93LNX
mqm       2648   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzmgr0 -m QM93LNX
mqm       2651   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzmuc0 -m QM93LNX
mqm       2666   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzmur0 -m QM93LNX
mqm       2681   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzmuf0 -m QM93LNX
mqm       2684   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqrrmfa -m QM93LNX -
t2332800 -s2592000 -p2592000 -g5184000 -c3600
mqm       2695   2634  0 11:38 ?       00:00:00 /opt/mqm/bin/amqzlaa0 -mQM93LNX -fip0
mqm       2813   2486  0 11:41 pts/1    00:00:00 amqsput Q1 QM93LNX
```

Because an amqsput connection is preventing the queue manager from terminating, then, only until the client amqsput terminates, then the queue manager will end.

One easy solution is to manually terminate amqsput, which will disconnect from the queue (OPPROCS will be 0) and the queue manager will terminate.

However, sometimes, it is not easy to identify which is the particular application that is preventing the shutdown, then it is OK to escalate the shutdown of the queue manager by issuing "-i" (for immediate).

Thus, in the same window or in another window, issue the following command, even though an "endmqm QMgrName" was issued already (but it is quiescing).

mqm@riggioni1.fyre.ibm.com: /home/mqm
$ **endmqm -i QM93LNX**
IBM MQ queue manager 'QM93LNX' ending.
IBM MQ queue manager 'QM93LNX' ended.

The "-i" flag will wait for in-transit activities to terminate, but because there are none at this time, then it will terminate the connection of amqsput and then endmqm will continue successfully to terminate the queue manager.

mqm@riggioni1.fyre.ibm.com: /var/mqm/errors
$ dspmq -m QM93LNX
QMNAME(QM93LNX) STATUS(Ended immediately)

Notice that the pending amqsput is terminated with rc 2009:

mqm@riggioni1.fyre.ibm.com: /home/mqm
$ amqsput Q1 QM93LNX
Sample AMQSPUT0 start
target queue is Q1

MQCLOSE ended with reason code 2009
Sample AMQSPUT0 end

$ mqrc 2009
2009 0x000007d9 MQRC_CONNECTION_BROKEN

++ Miscellaneous

+ General error log shows strmqm and endmqm

Notice that the "General" error log in /var/mqm/errors will contain the strmqm and endmqm entries

mqm@riggioni1.fyre.ibm.com: /var/mqm/errors
04/18/2023 11:38:16 AM - Process(2630.1) User(mqm) Program(strmqm)
Host(riggioni1.fyre.ibm.com) Installation(Installation1)
VRMF(9.3.2.0)
Time(2023-04-18T18:38:16.254Z)
CommentInsert1(strmqm)
CommentInsert2(strmqm QM93LNX)
AMQ6206I: Command strmqm was issued.
EXPLANATION:
Control command 'strmqm QM93LNX' was issued.
ACTION:
None.
----- amqxprma.c : 2750 --
04/18/2023 11:46:23 AM - Process(2823.1) User(mqm) Program(endmqm)
Host(riggioni1.fyre.ibm.com) Installation(Installation1)
VRMF(9.3.2.0)
Time(2023-04-18T18:46:23.093Z)
CommentInsert1(endmqm)
CommentInsert2(endmqm QM93LNX)
AMQ6206I: Command endmqm was issued.
EXPLANATION:
Control command 'endmqm QM93LNX' was issued.
ACTION:
None.
----- amqxprma.c : 2750 --
04/18/2023 11:51:08 AM - Process(2849.1) User(mqm) Program(endmqm)
Host(riggioni1.fyre.ibm.com) Installation(Installation1)
VRMF(9.3.2.0)
Time(2023-04-18T18:51:08.809Z)
CommentInsert1(endmqm)
CommentInsert2(endmqm -i QM93LNX)
AMQ6206I: Command endmqm was issued.
EXPLANATION:
Control command 'endmqm -i QM93LNX' was issued.
ACTION:
None.

++ General error log /var/mqm/errors shows AMQ9209E "Connection closed"

04/18/2023 11:35:28 AM - Process(16766.2) User(mqm) Program(amqsgetc)
Host(fosdinovo1.fyre.ibm.com) Installation(Installation1)
VRMF(9.1.0.13)
Time(2023-04-18T18:35:28.588Z)
RemoteHost(9.46.66.142(1415))
CommentInsert1(9.46.66.142(1415))
CommentInsert2(TCP/IP)
CommentInsert3(SYSTEM.DEF.SVRCONN)
AMQ9209E: Connection to host '9.46.66.142(1415)' for channel 'SYSTEM.DEF.SVRCONN' closed.
EXPLANATION:
An error occurred receiving data from '9.46.66.142(1415)' over TCP/IP. The connection to the remote host has unexpectedly terminated.
The channel name is 'SYSTEM.DEF.SVRCONN'; in some cases it cannot be determined and so is shown as '????'.
ACTION:

ACTION:
Tell the systems administrator.

+++ end +++