**MQ Production support**

**Monitoring**

* On-line status commands
* Event messages
* Off-line accounting and statistics

dspmq -o installation -o status

dspmqtrn -m BANCSCUSTQMGR

. /opt/mqm/bin/setmqinst -i -p /opt/mqm #set environment variable permanently

**Accounting and statistics messages:**

Accounting messages: used to record information about the MQI operations performed by WebSphere MQ applications.

Ex: MQI calls made using connections to a queue manager, number of MQI calls made using a connection to a MQGR.

Statistics messages: used to record information about the activities occurring in a WebSphere MQ system.

Ex: number of messages put on, and retrieved from, the queue, and the total number of bytes processed by a queue.

/opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t statistics -c NCB.SMARTHUB.01 –b

/opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER –b

/opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t accounting -q BANCS.TO.SMARTHUB.CUSTOMER –b

$ /opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER -b -d 2

**On-line status commands:**

DISLAY QMSTATUS

DISPLAY QMSTATUS ALL

DISPLAY QSTATUS(Q1) TYPE(QUEUE)

DISPLAY QSTATUS(Q1) TYPE(HANDLE)

dis chstatus(UAT.BESTCS.SWF.MQOUT) all

DISPLAY CONN(\*) WHERE(APPLTYPE EQ USER) TYPE(ALL) ALL \*Application status.

**Event messages**

**----**

$ ps -ef | grep lsr | grep -v grep \*List running listeners

$ ps -ef | grep runmqchi | grep -v grep \*channel initiator process

$ ps -ef | grep amqzx | grep -v grep \*execution controller process.

**Off-line accounting and statistics**

**MQ commands**

crtmqm Creates a queue manager

dltmqm Deletes a queue manager

dspmq Displays information about queue managers

dspmqini Displays queue manager initialization parameters

dspmqtrn Displays in-doubt and heuristically completed transactions

dspmqvar Displays global and queue manager environment variables

endmqm Stops a queue manager

rsvmqtrn Resolves in-doubt and heuristically completed transactions

runmqsc Runs MQSC commands

runswchl Switches transmission queue for cluster channel

setmqini Configures queue manager initialization parameters

setmqvar Configures global and queue manager environment variables

status Displays status information

strmqm Starts a queue manager

#strmqcsv - to start command server.

Display Number of message in queue.

echo "dis qs(BANCS.TO.SMARTHUB.CUSTOMER)" | /opt/mqm/bin/runmqsc BANCSCUSTQMGR

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc HLQ.EFTP.ETS.GBS | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)"

Display number of applications connected

bash-4.1$ echo 'display qstatus(SMARTHUB.TO.BANCS.CUSTOMER.ACK) ALL' | runmqsc BANCSCUSTQMGR | grep 'IPPROCS\|OPPROCS'

CURDEPTH(0) IPPROCS(1)

MSGAGE(0) OPPROCS(0)

IPPROCS : number of applications that are currently connected to the queue to get messages from the queue.

OPPROCS : number of applications that are currently connected to the queue to put messages on the queue.

PUT and GET count in a time interval

/opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER -b -s "2015-01-21 09.00.00" -e "2015-01-21 13.00.00"| egrep -wi 'IntervalStartDate|intervalstarttime|getcount|putcount'

Browse the message in queue

export MQSERVER='NCB.SMARTHUB.01/TCP/10.60.1.54(1414)'

cd /opt/mqm/samp/bin

./amqsbcgc BANCS.TO.SMARTHUB.CUSTOMER BANCSCUSTQMGR 2

Queue status attributes

$ display qstatus(SMARTHUB.TO.BANCS.CUSTOMER) monitor

**UNCOM**: his indicates whether there are any uncommitted changes (puts and gets) pending for the queue. If there are uncommitted changes pending, the value is Yes.

**QTIME**: The interval, in microseconds, between messages being put on the queue and then being destructively read. It includes any interval caused by a delay in committing by the putting application. a value based on recent activity over a short period of time, and a value based on activity over a longer period of time.

display qstatus(SMARTHUB.TO.BANCS.CUSTOMER) type(handle) ALL

**APPLTAG**: A string containing the tag of the application connected to the queue manager.

**PID**: The identifier of the process that opened the queue.

**TID**: The identifier of the thread within the application process that has opened the queue. An asterisk indicates that this queue was opened with a shared connection.

**APPLTYPE**: A string indicating the type of the application that is connected to the queue manager. Batch means an application using a batch connection; RRSBATCH means an RRS-coordinated application using a batch connection; CICS means a CICS transaction; IMS means an IMS transaction; CHINIT means a channel initiator; System means a queue manager; Usermeans a user application.

**APPLDESC**: A string containing a description of the application, where it is known, connected to the queue manager. If the application is not recognized by the queue manager the description returned will be all blanks. The application description will be filterable (e.g. with a WHERE clause) to allow an administrator to display only certain connections.

**BROWSE**: This indicates whether the handle is providing browse access to the queue. If the handle is providing browse access, the value is Yes;

**INQUIRE**: This indicates whether the handle is providing inquire access to the queue. If the handle is providing inquire access, the value is Yes

**INPUT:** This indicates whether the handle is providing input access to the queue. No means that the queue is not open for input;Shared means that the queue is open for shared input;Exclusive means that the queue is open for exclusive input.

**OUTPUT**: This indicates whether the handle is providing output access to the queue. If the handle is providing output access, the value is Yes;

SET: This indicates whether the handle is providing set access to the queue.

**USERID**: The user identifier that is associated with the handle.

**CHANNEL**: The name of the channel that owns the handle.

**Find a user’s level of permission on a queue.**

./dspmqaut –m BANCSCUSTQMGR –n BANCS.TO.SMARTHUB.CUSTOMER -t queue –p testuser1

./dspmqaut –m BANCSCUSTQMGR –t qmgr –p testuer1 \*If the user has connect permission to qmgr.

QMGR: BANCSCUSTQMGR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mqmadm   |  |  | | --- | --- | | DEV | 10.67.21.65 | | SIT | 10.60.1.54 | | UAT | 10.61.125.33 | | UAT | 10.61.125.34 | | PROD | 10.70.125.33 | | PROD | 10.70.125.34 | | DR | 10.80.125.33 | |

Queues

BANCS.TO.SMARTHUB.CUSTOMER

BANCS.TO.SMARTHUB.CUSTOMER.ACK

BANCS.TO.SMARTHUB.CUSTOMER.EXPIRY

SMARTHUB.DLQ

SMARTHUB.TO.BANCS.CUSTOMER

SMARTHUB.TO.BANCS.CUSTOMER.ACK

SMARTHUB.TO.BANCS.CUSTOMER.BATCH

SMARTHUB.TO.BANCS.CUSTOMER.BATCH.ACK

SMARTHUB.TO.BANCS.CUSTOMER.BATCH.EXPIRY

SMARTHUB.TO.BANCS.CUSTOMER.EXPIRY

Channels:

NCB.SMARTHUB.01

NCB.BANCS.01

Listener: BANCSCUSTQMGR.LSNR (1414

============================================================================

***QMGR status***

dspmq

strmqm /endmqm <queue manager>

dspmqtrn -m BANCSCUSTQMGR -a \* display queue manager transection details

# echo **'display QMSTATUS ALL'** | runmqsc BANCSCUSTQMGR

***Queue Status:***

echo 'display q(BANCS.TO.SMARTHUB.CUSTOMER)' | runmqsc BANCSCUSTQMGR \*queue properties

echo 'dis qs(SMARTHUB.TO.BANCS.CUSTOMER) **type(handle)** all' | runmqsc BANCSCUSTQMGR

echo 'clear QL(SMARTHUB.DLQ)' | runmqsc BANCSCUSTQMGR

**Monitoring of Queue:**

# echo 'ALTER qLOCAL(BANCS.TO.SMARTHUB.CUSTOMER) MONQ(HIGH)' | runmqsc BANCSCUSTQMGR (Enable queue monitoring)

# echo 'display qstatus(BANCS.TO.SMARTHUB.CUSTOMER.ACK) ALL' | runmqsc BANCSCUSTQMGR

CURDEPTH

IPPROCS: The number of handles that are currently open for input for the queue

OPPROCS: This is the number of handles that are currently open for output for the queue.

LGETDATE

LGETTIME

LPUTDATE

LPUTTIME

MSGAGE: Age, in seconds, of the oldest message on the queue.

QTIME: Interval, in microseconds, between messages being put on the queue and then being destructively read.

It includes any interval caused by a delay in committing by the putting application.

Two values are displayed:

- A value based on recent activity over a short period of time.

- A value based on activity over a longer period of time."

***Queue Handle status:***

# echo 'dis qs(BANCS.TO.SMARTHUB.CUSTOMER.ACK) type(handle) ALL' | runmqsc BANCSCUSTQMGR \*connected IP, PID, Channel, Application name, Appln tag.

APPLDESC: A string containing a description of the application connected to the queue manager

APPLTAG: A string containing the tag of the application connected to the queue manager.

APPLTYPE: Type of the application connected(BATCH, USER, CICS,IMS, CHINIT)

ASTATE: The state of the asynchronous consumer on this queue.

CHANNEL: The name of the channel that owns the handle.

HSTATE: Whether an API call is in progress.

***Channel Status :***

echo 'display chl(NCB.BANCS.01)' | runmqsc BANCSCUSTQMGR \*Channel properties

echo 'display chstatus(NCB.BANCS.01)' | runmqsc BANCSCUSTQMGR \*current status

***Application Status:***

# echo 'DISPLAY CONN(\*) WHERE(APPLTYPE EQ USER) TYPE(ALL)' | runmqsc BANCSCUSTQMGR

DISPLAY CONN provides information about applications connected to the QMGR and the handles they have open.

Who, what, client IP, Application name, connection time, connected Queue.

***Listener status:***

echo 'display LISTENER(BANCSCUSTQMGR.LSNR)' | runmqsc BANCSCUSTQMGR \* properties

echo 'display LSSTATUS('BANCSCUSTQMGR.LSNR')' | runmqsc BANCSCUSTQMGR \* current status

echo 'START LISTENER(BANCSCUSTQMGR.LSNR)' | runmqsc BANCSCUSTQMGR

echo 'STOP LISTENER(BANCSCUSTQMGR.LSNR)' | runmqsc BANCSCUSTQMGR

$ runmqlsr -t tcp -m QM\_TEST -p 1414

$ endmqlsr -m BANCSCUSTQMGR \* To stop listener

$ runmqlsr -t tcp -p <port> -m <queue manager name>

**Get and put to queue**

export MQSERVER='NCB.SMARTHUB.01/TCP/10.70.125.33(1414)'

./amqsputc BANCS.TO.SMARTHUB.CUSTOMER BANCSCUSTQMGR 16 BANCSCUSTQMGR \*writing queue

./amqsbcgc BANCS.TO.SMARTHUB.CUSTOMER BANCSCUSTQMGR 2 \*Reading queue

**Security:**

**echo 'dis chlauth(NCB.SMARTHUB.01)' | runmqsc BANCSCUSTQMGR | grep -w ADDRESS**

**echo 'dis chlauth(NCB.BANCS.01)' | runmqsc BANCSCUSTQMGR | grep -w ADDRESS**

echo 'dis qmgr chlauth' | runmqsc BANCSCUSTQMGR \*display QMGR authentication.

echo 'alter qmgr chlauth(enabled/disabled)' | runmqsc BANCSCUSTQMGR \*To enable/ disable authentication

echo 'dis chlauth(\*)' | runmqsc BANCSCUSTQMGR \*display ALL chlauth records.

# echo 'DISPLAY QMGR AUTHOREV' | runmqsc BANCSCUSTQMGR # view security trail status

# echo 'ALTER QMGR AUTHOREV (ENABLED)' | runmqsc BANCSCUSTQMGR # Enable security trail on QMGR.

set chlauth(NCB.SMARTHUB.01) type(addressmap) address('smarthubapp\*') usersrc(CHANNEL) chckclnt(ASQMGR)

set chlauth(NCB.BANCS.01) type(addressmap) address('rhlux3166.alahli.com') usersrc(CHANNEL) chckclnt(ASQMGR)

set chlauth(NCB.SMARTHUB.01) type(addressmap) address('10.155.20.179') usersrc(CHANNEL) chckclnt(ASQMGR)

refresh security type(connauth)

***Process monitor***

$ ps -ef | grep lsr | grep -v grep \*List running listeners

$ ps -ef | grep runmqchi | grep -v grep \*channel initiator process

$ ps -ef | grep amqzx | grep -v grep \*execution controller process.

***Log***

/var/mqm/errors

/var/mqm/qmgrs/<queueManagerName>/errors

***Statistics Messages***

Collects information about WebSphere MQ resources. Queue Manager, queue, channel.

# echo 'display ql(SMARTHUB.TO.BANCS.CUSTOMER) STATQ' | runmqsc BANCSCUSTQMGR

: ALTER ql(SMARTHUB.TO.BANCS.CUSTOMER) STATQ(ON)

Or

: ALTER QMGR STATMQI(ON) STATQ(ON) STATCHL(HIGH) STATACLS(QMGR)

: ALTER QLOCAL(q-name) STATQ(QMGR)

#cd /opt/mqm/samp/bin

# ./amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER

# ./amqsmon -m BANCSCUSTQMGR -t statistics -w 10 -q SMARTHUB.TO.BANCS.CUSTOMER (w=wait in sec)

# ./amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER –b -s "2012-12-22 16.36.00" -e "2012-12-22 16.36.00"

#./amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER -s "2012-12-22" -e "2012-12-22"

/opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER -b -s "2015-01-18 00.00.00" -e "2015-01-19 00.00.00" | egrep -wi 'intervalstarttime|getcount|putcount'

# /opt/mqm/samp/bin/amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER -b | egrep -wi 'QueueName|IntervalStartDate|intervalstarttime|getcount|putcount' | tail -5

BrowseFailCount should reflect the number of unsuccessful non-destructive MQGET requests

GetFailCount should reflect the number of unsuccessful destructive get requests.

***Accounting Messages***

Collects information about the applications which connect to the Queue Manager

# echo 'display ql(SMARTHUB.TO.BANCS.CUSTOMER) ACCTQ' | runmqsc BANCSCUSTQMGR

echo 'ALTER qLOCAL(SMARTHUB.TO.BANCS.CUSTOMER) ACCTQ(ON)' | runmqsc BANCSCUSTQMGR

or

: ALTER QMGR ACCTMQI(ON) ACCTQ(ON) ACCTCHL(ON)

: ALTER QLOCAL(q-name) ACCTQ(QMGR)

# echo 'ALTER ql(SMARTHUB.TO.BANCS.CUSTOMER) STATQ(ON)' | runmqsc BANCSCUSTQMGR

#cd /opt/mqm/samp/bin

# ./amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER

# ./amqsmon -m BANCSCUSTQMGR -t statistics -w 10 -q SMARTHUB.TO.BANCS.CUSTOMER (w=wait in sec)

# ./amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER -s "2012-12-22 16.36.00" -e "2012-12-22 16.36.00"

#./amqsmon -m BANCSCUSTQMGR -t statistics -q SMARTHUB.TO.BANCS.CUSTOMER -s "2012-12-22" -e "2012-12-22"

# echo "reset qmgr type(statistics)"|runmqsc BANCSCUSTQMGR

# ./amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER \* command displays all queue statistics messages for queue LOCALQ on queue manager

#./amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER |grep -i -e queuename -e putcount -e getcount -e putbyte -e getbyte

# echo "reset qmgr type(statistics)"|runmqsc BANCSCUSTQMGR

# ./amqsmon -m BANCSCUSTQMGR -t statistics -q BANCS.TO.SMARTHUB.CUSTOMER -s "2015-01-03 00.00.00" | grep -i -e putcount -e getcount

***License:***

File location

/opt/mqm/lib/

* amq**t**cert.lic – is a trial license
* amq**b**cert.lic – is a beta license
* amq**p**cert.lic – is a full production license

./ [setmqprd](http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/topic/com.ibm.mq.amqzag.doc/zmqprd.htm) \* command to upgrade license.

$MQ\_INSTALLATION\_PATH/bin/setmqprd /MediaRoot/licenses/amqpcert.lic

# dspmqver

**Queue Alerts:**

If any message is >3600s old in any queue.

echo "dis qs(SMARTHUB\*) where( MSGAGE gt 3600)" | runmqsc BANCSCUSTQMGR | grep 'QUEUE\| MSGAGE' | mail -s MessageAge 151730@alahli.com,n.jain@alahli.com,r.alzahrani@alahli.com,s.bardhan@alahli.com,h.ali2@alahli.com,s.ahmed3@alahli.com

If any queue has >500 messages.

echo "dis q(SMARTHUB\*) where(curdepth gt 500)" | runmqsc BANCSCUSTQMGR | grep 'QUEUE\| CURDEPTH' | mail -s QueueDepth 151730@alahli.com,n.jain@alahli.com,r.alzahrani@alahli.com,s.bardhan@alahli.com,h.ali2@alahli.com,s.ahmed3@alahli.com

If SMARTHUB.DLQ has >0 message.

echo "dis qs(SMARTHUB.DLQ) where(curdepth gt 0)" | runmqsc BANCSCUSTQMGR | grep 'QUEUE\| CURDEPTH' | mail -s DeadLetterQueue 151730@alahli.com,n.jain@alahli.com,r.alzahrani@alahli.com,s.bardhan@alahli.com,h.ali2@alahli.com,s.ahmed3@alahli.com

If an EXPIRY queue has >0 message

echo "dis qs(BANCS.TO.SMARTHUB.CUSTOMER.EXPIRY) where(curdepth gt 0)" | runmqsc BANCSCUSTQMGR | grep 'QUEUE\| CURDEPTH' | mail -s BANCS.TO.SMARTHUB.CUSTOMER.EXPIRY 151730@alahli.com,n.jain@alahli.com,r.alzahrani@alahli.com,s.bardhan@alahli.com,h.ali2@alahli.com,s.ahmed3@alahli.com

echo "dis qs( SMARTHUB.TO.BANCS.CUSTOMER.BATCH.EXPIRY ) where(curdepth gt 0)" | runmqsc BANCSCUSTQMGR | grep 'QUEUE\| CURDEPTH' | mail -s SMARTHUB.TO.BANCS.CUSTOMER.BATCH.EXPIRY 151730@alahli.com,n.jain@alahli.com,r.alzahrani@alahli.com,s.bardhan@alahli.com,h.ali2@alahli.com,s.ahmed3@alahli.com

echo "dis qs( SMARTHUB.TO.BANCS.CUSTOMER.EXPIRY ) where(curdepth gt 0)" | runmqsc BANCSCUSTQMGR | grep 'QUEUE\| CURDEPTH' | mail -s SMARTHUB.TO.BANCS.CUSTOMER.EXPIRY 151730@alahli.com,n.jain@alahli.com,r.alzahrani@alahli.com,s.bardhan@alahli.com,h.ali2@alahli.com,s.ahmed3@alahli.com

check if there is any consumer to the queue on Ibm Mq

echo "dis qs(\*) type(handle) all" | runmqsc BANCSCUSTQMGR | grep 'INPUT\|QUEUE'

echo "dis qs(SMARTHUB\*) type(handle) all" | runmqsc BANCSCUSTQMGR | grep 'OUTPUT\|QUEUE'

**SupportPac MH04 to effectively monitor queue depths.**

**Test connection**

-bash-4.1$ export MQSERVER='NETREVEAL.TO.CORDYS.01/TCP/10.60.1.54(1415)'

-bash-4.1$ export MQSAMP\_USER\_ID=userrw

-bash-4.1$ cd /opt/mqm/samp/bin/

-bash-4.1$ ./amqsputc SWIFT.TO.NETREVEAL NCBSSS 16 NCBSSS

To change Max queue depth

alter ql('NETREVEAL.TO.MONEYGRAM.FINAL') MAXDEPTH(5000)

**30. Mount GPFS cluster FS**

# mmlsmount all

**31. How to find how many messages are passed through the particular channel**

#runmqsc<QMGR NAME>

DISPLAY CHSTATUS(NETREV.TO.CORDYS.01) ALL

Attribute called MSGS tells how many messages passed through that particular channel instance.

**32. Monitoring using the dmpmqmsg utility**

#dmpmqmsg –m QM1 –i Q1 –f c:\myfile

#dmpmqmsg -m NCBSSS -i NETREVEAL.TO.QUICKPAY.INITIAL -f /var/mqm/test1

\*Takes a copy of the messages from the queue and saves them in the file specified.

dmpmqmsg –m QM1 –i Q1 –f c:\myfile%n

$ dmpmqmsg -m NCBSSS -i NETREVEAL.TO.QUICKPAY.INITIAL -f /var/mqm/test%n

In this mode each message is written to a new file, nloads the queue to files, myfile1, myfile2, myfile3, and so on.

# dmpmqmsg –m QM1 –o Q1 –f c:\myfile%n

\*To reload a queue with the messages you saved in.

#dmpmqmsg –m QM1 –i Q1 –o Q2

$ dmpmqmsg -m NCBSSS -i NETREVEAL.TO.QUICKPAY.INITIAL -o SSS.DLQ

allows the messages from one queue to be copied to another queue.

dmpmqmsg –m QM1 –i Q1 –o Q2 -r#10

$ dmpmqmsg -m NCBSSS -i NETREVEAL.TO.QUICKPAY.INITIAL -o SSS.DLQ -r#6

Copy the first 10 messages from one queue to another queue

dmpmqmsg –m QM1 –I Q1 –o Q2

Move the messages from one queue to another queue

dmpmqmsg –m QM1 –I Q1 –o Q2 -T1440

Move messages older than one day from one queue to another queue

dmpmqmsg –m QM1 –i Q1 –f stdout –dT

$ dmpmqmsg -m NCBSSS -i NETREVEAL.TO.QUICKPAY.INITIAL -f stdout -dT

Display the ages of messages currently on a queue.

Best way to get message ID and age of each message.

**33. Queue Monitoring scripts**

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc *BANCSCUSTQMGR* | grep -E "(QUEUE\(|CURDEPTH\()" |perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)" | perl -pe 's/^[\s]+$//g' | perl -pe 's/TYPE\(QLOCAL\)//g' | perl -pe "s/^/IP($IP)\tQM\($QM)\t/g"

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc *BANCSCUSTQMGR* | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)" | sed -e 's/TYPE(QLOCAL)//g'

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc *BANCSCUSTQMGR* | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)" | sed -e 's/TYPE(QLOCAL)//g' | grep -v '(0)'

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc NCBSSS | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)"

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc TCSBESS | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)"

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc RATEPAY | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)"

echo "DISPLAY QL(\*) CURDEPTH" | runmqsc PAYMENTSQMGR | grep -E "(QUEUE\(|CURDEPTH\()" | perl -pe 's/\n//g' | perl -pe 's/QUEUE\(/\nQUEUE\(/g' | grep -vE "QUEUE\((SYSTEM|AMQ)"

**34. Using WebSphere MQ Explorer as a read-only viewer**

Firstly, you need permission to connect to the queue manager:

setmqaut -m YOUR\_QUEUE\_MANAGER -t qmgr -p YOUR\_USER\_NAME +connect +inq +dsp

Next, you need to give permission to the queues that WMQ Explorer will need:

setmqaut -m YOUR\_QUEUE\_MANAGER -t q -n SYSTEM.DEFAULT.MODEL.QUEUE -p YOUR\_USER\_NAME +get +browse +inq

setmqaut -m YOUR\_QUEUE\_MANAGER -t q -n SYSTEM.ADMIN.COMMAND.QUEUE -p YOUR\_USER\_NAME +get +browse +inq +put

setmqaut -m YOUR\_QUEUE\_MANAGER -t q -n SYSTEM.MQEXPLORER.REPLY.MODEL -p YOUR\_USER\_NAME **+inq** +browse +get +dsp

-----Done---

To give read only view to all queues

setmqaut -m YOUR\_QUEUE\_MANAGER -t channel -n '\*\*' -p YOUR\_USER\_NAME +dsp

setmqaut -m YOUR\_QUEUE\_MANAGER -t q -n '\*\*' -p YOUR\_USER\_NAME +dsp +inq +browse