MQ Replay a Persistent message from the log - Day 2 Lab #5


An
IBM
Washington
Systems
Center
Wildfire Workshop

Lab Objective

This lab is to demonstrate using the provided utilities to replay a persistent message from the queue manager log. It will employ CSQUTIL to create a queue, use OEMPUT to put a single persistent message, use the MQ Explorer to look at the message, OEMPUT to get the message from the queue, CSQ1LOGP to extract the message based on the pageset used, CSQ4LOGS to replay the message, and the same module to do an activity summary.

Lab Steps

1) In the TEAMXX.MQPERF.LOG.JCL PDS (where the TEAMXX is your TEAM ID), select the DEFQLOG member. This job defines two queues that will be used in the jobs that follow.

```
EDIT       TEAMXX.MQPERF.LOG.JCL(DEFQLOG) - 01.03                Columns 00001 00080
Command ===>                                                        Scroll ===> CSR
****** ********************************* Top of Data ***********************************
000001 //++TEAMXX++DS JOB (????,????),'DEFINES',NOTIFY=????????                 00010000
000002 /*JOBPARM SYSAFF=(++LPAR++)
000003 //*
000004 //* THIS CSQUTIL TASK DEFINES THE QUEUES USED IN THE LAB EXERCISES
000005 //*
000006 //* MAKE THE FOLLOWING CHANGES TO THE JCL:
000007 //*  1) CHANGE ++TEAMXX++ TO YOUR TEAM ID
000008 //*  2) CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
000009 //*  3) CHANGE ++QMGR++ TO THE QMGR ON YOUR WORKSHEET
000010 //*  4) CHANGE ++MQHLQ++ TO THE MQ HIGH LEVEL QUALIFIER
000011 //*  5) CHANGE ++STGCLAS++ TO YOUR TEAM NUMBER
000012 //*
```

2) Using a change all command, alter the '++' variables as listed in the JCL comments:

   CHANGE ++TEAMXX++ TO YOUR TEAM ID
   CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
   CHANGE ++QMGR++ TO THE QMGR ON YOUR WORKSHEET
   CHANGE ++MQHLQ++ TO THE MQ HIGH LEVEL QUALIFIER
   CHANGE ++STGCLAS++ TO YOUR TEAM NUMBER

> Tech Tip: the change all command has the format of 'C' (for change) , the value to be replaced , the replacement value, and the word ALL. If either the original value or the replacement contains a space, they can be enclosed in quotes. As example, changing the ++ variable to the team id is shown.
> C ++TEAMXX++ TEAM20 ALL

NOTE: The MQHLQ is MQ910. The STGCLAS may be different from the pageset ID, which will be

used later to isolate the message(s) to extract and replay.

3) After the changes the statements should looks something like this (TEAM20 used as an example)

   CHANGE TEAM20 TO YOUR TEAM ID
   CHANGE MPX2 TO THE LPAR ON YOUR WORKSHEET
   CHANGE QML2 TO THE QMGR ON YOUR WORKSHEET
   CHANGE MQ910 TO THE MQ HIGH LEVEL QUALIFIER
   CHANGE 20 TO YOUR TEAM NUMBER

4) Save and submit the job.
5) Navigate to SDSF.ST to review the results.

6) Select the job, using the question mark to display the list of files:

```
SDSF JOB DATA SET DISPLAY - JOB TEAM20DS (JOB07993)
COMMAND INPUT ===>
NP   DDNAME   StepName ProcStep DSID Owner    C Dest
     JESMSGLG JES2                 2 ELKINSC  S LOCAL
     JESJCL   JES2                 3 ELKINSC  S LOCAL
     JESYSMSG JES2                 4 ELKINSC  S LOCAL
s    SYSPRINT DEFQS              103 ELKINSC  S LOCAL
```

7) Select it as shown. The output should indicate success – this is important to check, as there are times when the job will receive a zero return code, but the queue has not actually been created. The output should look like this:

```
CSQU000I CSQUTIL IBM MQ for z/OS V9.1.0
CSQU001I CSQUTIL Queue Manager Utility - 2019-03-25 15:36:15
COMMAND
CSQU127I  Executing COMMAND using input from CSQUCMD data set
CSQU120I  Connecting to QML2
CSQU121I  Connected to queue manager QML2
CSQU055I  Target queue manager is QML2
 DEFINE QLOCAL('TEAM20C.PERSIST.COPYIN') QSGDISP(QMGR)              -
        STGCLASS(STGCLS20) DEFPSIST(YES)                            -
        DEFSOPT(SHARED) MONQ(HIGH)                                  -
        SHARE REPLACE
CSQN205I   COUNT=       2, RETURN=00000000, REASON=00000000
```

8) Return to the JCL PDS and select the LOGPUT member.
9) Using a change all command, alter the '++' variables as listed in the JCL comments:

   CHANGE ++TEAMXX++ TO YOUR TEAM ID
   CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
   CHANGE ++QMGR++ TO THE QMGR ON YOUR WORKSHEET
   CHANGE ++MQHLQ++ TO THE MQ HIGH LEVEL QUALIFIER

10) After the changes the statements should looks something like this (TEAM20 used as an example)

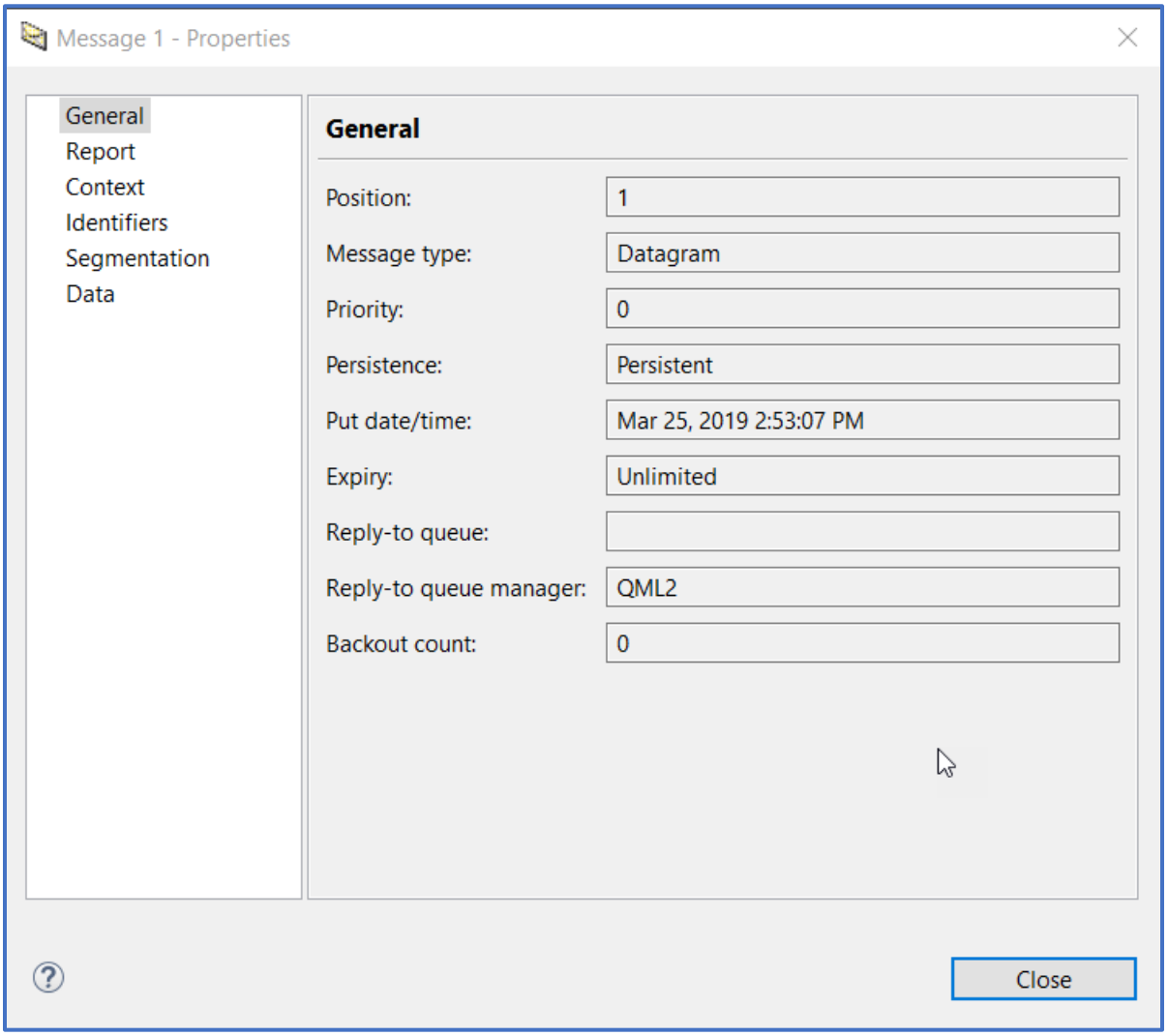
   CHANGE TEAM20C TO YOUR TEAM ID
   CHANGE MPX2 TO THE LPAR ON YOUR WORKSHEET
   CHANGE QML2 TO THE QMGR ON YOUR WORKSHEET
   CHANGE MQ910 TO THE MQ HIGH LEVEL QUALIFIER

11) Save and submit the JCL.
12) Navigating to the output, using the SDSF.ST option, the SYSPRINT output should look something like the example shown here:

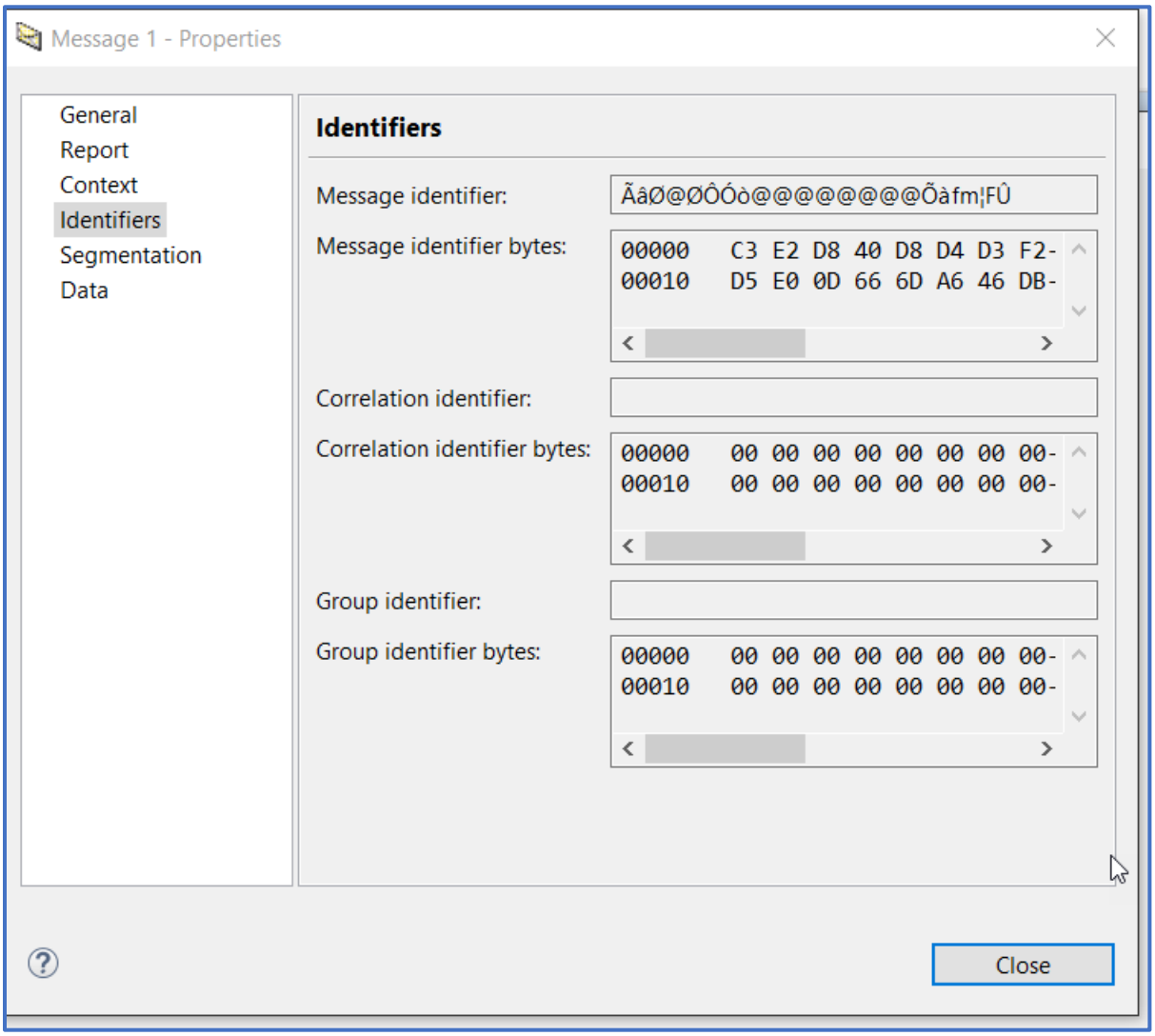
```
parm: -mQML2 -cgpc -put1 -qTEAM20C.PERSIST.COPYIN -s80 -n1 -l1 -fileDD:MIN -p -crlf
Message data generate from each file line
Message file -FILE: DD:MIN open mode:rb, type=record
bytes read from msg file  80
reply size 104857600
OEMPUT about to MQCONN to QMgr QML2.
CPU type 0000012827
Date Time 2019/03/25 19:53:07.
Using MQPUT1
Entering PUT only loops...
Preload the queue with 0 messages...
  Message size        : 80
  Reply size          : 104857600
  Message persistence : PERSISTENT
  Messages per loop   : 1
  Total messages      : 1
  Syncpoints Get 1,  Put 1, Commit in syncpoint
Starting loop at 2019-03-25 19:53:07.313433
```

13) Open the MQ Explorer, if the you do not see the queue managers QML1 or QML2 please go to the Appendix to see how to add them to the list of available queue managers.
14) Browse the messages on the queue. To browse the messages, right click on the queue name and select Browse Messages.

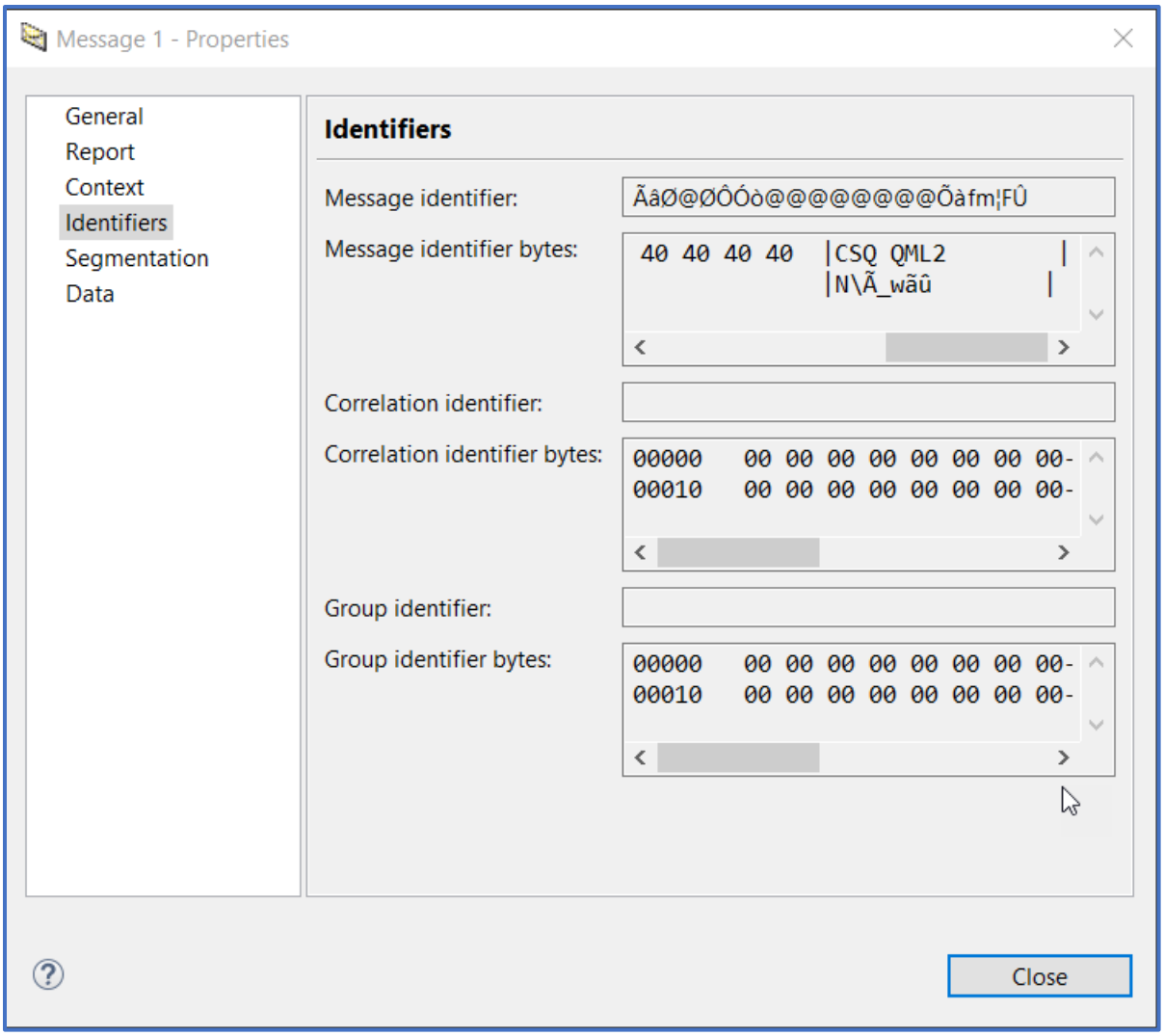
15) Right click on the message and select 'Properties' . This should open the following panel:



16) Select the ‘Identifiers’ tab and look at the message ID



17) Using the scroll, move to the right displaying the remainder of the message identifier bytes.



18) Close the panel.
19) Returning to the JCL PDS, select the LOGGET member and make changes to the '++' variables like the changes made to the LOGPUT member.
20) Save and submit the LOGGET job. Note that this task will take slightly more than a minute to complete, as it defaults to a get wait of 60 seconds. In addition it will return a 2033 reason code.

21) Navigating to the output, using the SDSF.ST option, the SYSPRINT output should look something like the example shown here:

```
parm: -mQML2 -cgpc -put1 -rTEAM20C.PERSIST.COPYIN -s100 -n1 -l5 -fileDD:MIN -p -crlf
Message data generate from each file line
Message file -FILE: DD:MIN open mode:rb, type=record
bytes read from msg file  80
reply size 104857600
OEMPUT about to MQCONN to QMgr QML2.
CPU type 0000012827
Date Time 2019/03/25 20:01:09.
Using MQPUT1
Entering GET loops (MQGET_Wait=60 seconds)...
  Messages per loop   : 5
  Total messages      : 1
  Syncpoints Get 5,  Put 5, Commit in syncpoint
  MQGET replies by    : Any message
Starting loop at 2019-03-25 20:01:09.658494
OEMPUT: MQGET failed cc=2, rc=2033  MQRC_NO_MSG_AVAILABLE
MQGET Rcode 2033
```

22) Note that the total messages retrieved is 1. Returning to the MQ Explorer, the current queue depth should now be zero.
23) At this point, you have now successfully put and gotten a persistent message.
24) Returning to the JCL PDS, select the LOGEXTR1 member. This is an execution of the CSQ1LOGP task. Make the global changes to the '++' variables, with the exception of the PAGESET and active log variables.

    CHANGE ++PAGESET++ TO THE PAGESET NUMBER USED FOR THE QUEUE
    CHANGE ++LOGNAME++ TO THE ACTIVE LOG WHERE THE MESSAGE WAS PUT

25) To get the pageset number:

a. From the MQ Explorer, display the storage classes. It should look something like this:

Storage class name	Page set ID	QSG disposition
DEFAULT	4	Queue manager
NODEFINE	4	Queue manager
REMOTE	3	Queue manager
STGCLAS0	42	Queue manager
STGCLS00	0	Queue manager
STGCLS01	1	Queue manager
STGCLS02	2	Queue manager
STGCLS03	3	Queue manager
STGCLS04	4	Queue manager
STGCLS05	21	Queue manager
STGCLS06	41	Queue manager
STGCLS07	42	Queue manager
STGCLS08	43	Queue manager
STGCLS09	51	Queue manager
STGCLS10	10	Queue manager
STGCLS11	11	Queue manager
STGCLS12	12	Queue manager
STGCLS13	13	Queue manager
STGCLS14	14	Queue manager
STGCLS15	15	Queue manager
STGCLS16	16	Queue manager
STGCLS17	17	Queue manager
STGCLS18	18	Queue manager
STGCLS19	19	Queue manager
STGCLS20	20	Queue manager

b. Match the name of the storage class used to the pageset ID. In the example for the lab document, the pageset associated with STGCLS09 is '51'.

c. Change the all instances of the ++PAGESET++ variable to the pageset from this list.

26) To get the name of the active log data set, navigate to the SDSF.DA panel and alter the prefix to display the queue manager and channel initiator. The command to do that is PREFIX QML* and the output should look something like this (example is for the MPX2 LPAR):

```
SDSF DA MPX2     MPX2     PAG  0  CPU   3              LINE 1-4 (4)
COMMAND INPUT ===>                                        SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     QML2CHIN QML2CHIN PROCSTEP STC07411 MQUSER     NS  FE 4584   0.00   0.00
     QML2MSTR QML2MSTR PROCSTEP STC07410 MQUSER     NS  FE 137T   0.00   0.00
     QML4CHIN QML4CHIN PROCSTEP STC07413 MQUSER     NS  FE 4507   0.00   0.00
     QML4MSTR QML4MSTR PROCSTEP STC07412 MQUSER     NS  FE 135T   0.00   0.00
```

27) Expand the MSTR output for your primary queue manager and select the JESMSGLG output. The results should look something like what is shown:

```
 SDSF OUTPUT DISPLAY QML2MSTR STC07410  DSID     2 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA ***********************************
                     J E S 2  J O B  L O G  --  S Y S T E M  M P X 2  --  N O D

12.57.58 STC07410 ---- SUNDAY,     17 FEB 2019 ----
12.57.58 STC07410  IEF695I START QML2MSTR WITH JOBNAME QML2MSTR IS ASSIGNED TO U
12.57.58 STC07410  $HASP373 QML2MSTR STARTED
12.57.58 STC07410  IEF403I QML2MSTR - STARTED - TIME=12.57.58
12.57.58 STC07410  CSQY000I QML2 IBM MQ for z/OS V9.0.0 LTSR
12.57.58 STC07410  CSQY001I QML2 QUEUE MANAGER STARTING, USING PARAMETER MODULE
12.57.58 STC07410  CSQ3111I QML2 CSQYSCMD - EARLY PROCESSING PROGRAM IS V9.1.1 L
   962             008-000
12.57.58 STC07410  CSQY100I QML2 SYSTEM parameters ...
```

28) Navigate to the bottom of the output using the 'BOT' command.
29) Enter the command to display the active log, replacing the '+cpf' with the queue manager name:

    +cpf DISPLAY LOG

30) At the end of the Display log report, the current active log is given. In the case of QML2 at the time this was written, this looks as follows:

```
20.43.48 STC07410  CSQJ370I QML2 LOG status report ...   081
   081             Copy %Full PPRC DSName
   081              1      1  NO   WMQ900.QML2.LOGCOPY1.DS003
   081              2              Inactive
   081             Restarted at 2019-02-17 12:57:58 using RBA=00000000007B3000
   081             Latest RBA=00000000008BFF4B
   081             Offload task is AVAILABLE
   081             Full logs to offload - 0 of 4
20.43.48 STC07410  CSQ9022I QML2 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION
```

31) The last node of the current log name is used to replace the '++LOGNAME++' variable. In this case the value is DS003.
32) Change all instances of the ++LOGNAME++ to the value above. Save and submit the LOGEXTR1 job. This will extract all the updates made to the pageset. Note that in a normal production environment this could include many messages for many queues. The documentation for this may be found here:

    https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.1.0/com.ibm.mq.ref.adm.doc/q088970_.htm

33) The start of the output from this job should look as follows:

```
CSQ1000I CSQ1LOGP IBM MQ for z/OS V9.1.0

CSQ1100I CSQ1LOGP LOG PRINT UTILITY - 2019-04-03 17:08:45

=================================================================
PAGESET(51)
EXTRACT(YES)

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

CSQ1102I SEARCH CRITERIA

SUMMARY(NO)
ALL URIDS
ALL RMS
PAGESET(00000033)
EXTRACT(YES)
```

NOTE: The pageset in the 'SEARCH CRITERIA' section is the hex value for the pageset.

34) Scrolling thru the output, you should be able to see the message much like what is shown here:

```
  SUBTYPE( INSERT )  RECORD LENGTH(01F6)
  LRSN(D5E00D666DAA)                                                                          15:53:07.314336 20190325
****  021E0032 0400000C C9C00000 008BE034 0000008B E0CD9020 D5E00D66 6DAA0003  *        I               N   _
0000  00000014 00000201 01F60900 00000000 00030000 000A0000 00005000 00020101  *         6                &
0020  A6000000 01000000 01600000 02010046 00000000 00000000 00000000 00000000  *w        -
0040  00000000 00000000 00000000 00000160 D4150000 0001D5E0 0D666DA6 46DB0000  *              -M     N   _w
0060  0050FFFF FFFF0000 0000D4C4 40400000 00010000 00000000 0008FFFF FFFF0000  * &        MD
0080  00000000 03110000 01F4D4D8 E2E3D940 40400000 00000000 0001C3E2 D840D8D4  *         4MQSTR           CSQ QM
00A0  D3F24040 40404040 4040D5E0 0D666DA6 46DB0000 00000000 00000000 00000000  *L2        N   _w
00C0  00000000 00000000 00000000 00004040 40404040 40404040 40404040 40404040  *
00E0  40404040 40404040 40404040 40404040 40404040 40404040 40404040 4040D8D4  *                                QM
0100  D3F24040 40404040 40404040 40404040 40404040 40404040 40404040 40404040  *L2
0120  40404040 40404040 40404040 4040C5D3 D2C9D5E2 C3404040 404008D5 D6C9C5C6  *              ELKINSC      NOIEF
0140  E4D1C900 00000000 00000000 00000000 00000000 00000000 00004040 40404040  *UJI
0160  40404040 40404040 40404040 40404040 40404040 40404040 40400000 0002E3C5  *                                TE
0180  C1D4F2F0 D7404040 40404040 40404040 40404040 40404040 4040F2F0 F1F9F0F3  *AM20P                        201903
01A0  F2F5F1F9 F5F3F0F7 F3F14040 4040E388 89A24089 A2409596 A3408195 40859485  *2519530731    This is not an eme
01C0  99878595 83A84040 40404040 40404040 40404040 40404040 40404040 40404040  *rgency
01E0  40404040 40404040 40404040 40404040 40404040 4040F0F0 F0F1F0F0 F0F0      *                        00010000
```

35) Return to the JCL PDS and select the LOGJ member. This executes the sample program CSQ4LOGS to replay the message(s) from the file created by the log extract process.

36) Alter the '++' variables as done for the previous jobs. Save and submit the job.
37) The output, in SYSPRINT, gives a description of the actions taken. In this case, the start should look something like this:

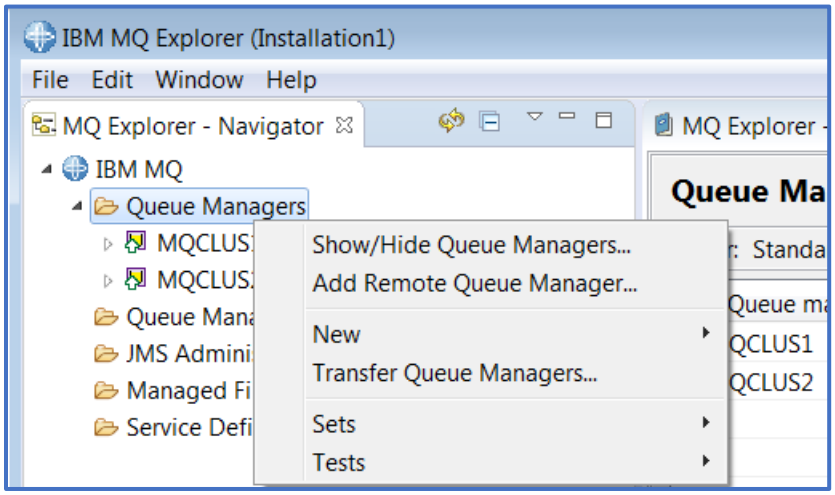
```
CSQ4LOGS - Sample log extract replay program
parm 1, QML2
parm 2, REPLAY
Msgs to non-system queues will be replayed
Opening input file

==> New Unit of Recovery started <==
Unit of recovery ID ....' (hex) '00000000008BE034' : (disp) '......\.'
Correlator ID...........' (hex) '00000000000000000000000' : (disp) '............'
Authorisation ID........' (hex) 'C5D3D2C9D5E2C340' : (disp) 'ELKINSC '
Thread start TOD value..' (hex) 'D5E00D666DA8FBDB' : (disp) 'N\..._y..'
Resource identifier.....' (hex) '4040404040404040' : (disp) '        '
Connection type.........' (hex) 'C2C1E3C3C8404040' : (disp) 'BATCH   '
Connection identifier...' (hex) 'E3C5C1D4F2F0D740' : (disp) 'TEAM20P '
Connect to QML2 complete, compcode 0 Reason 0
MQPUT1 - Length=80,Queue=TEAM20C.PERSIST.COPYIN                          CC=0, RC=0
==> Summary of UR activity       <==
==> Message puts : 00000001, ==> Message gets : 00000000, ==> Message expired : 00000000, ==> Object defines : 00000000, ==> Object
alters : 00000000.
==> End summary of UR activity   <==
```

38) In addition, the queue depth should be back to 1 and browsing the queue should show the restored message.

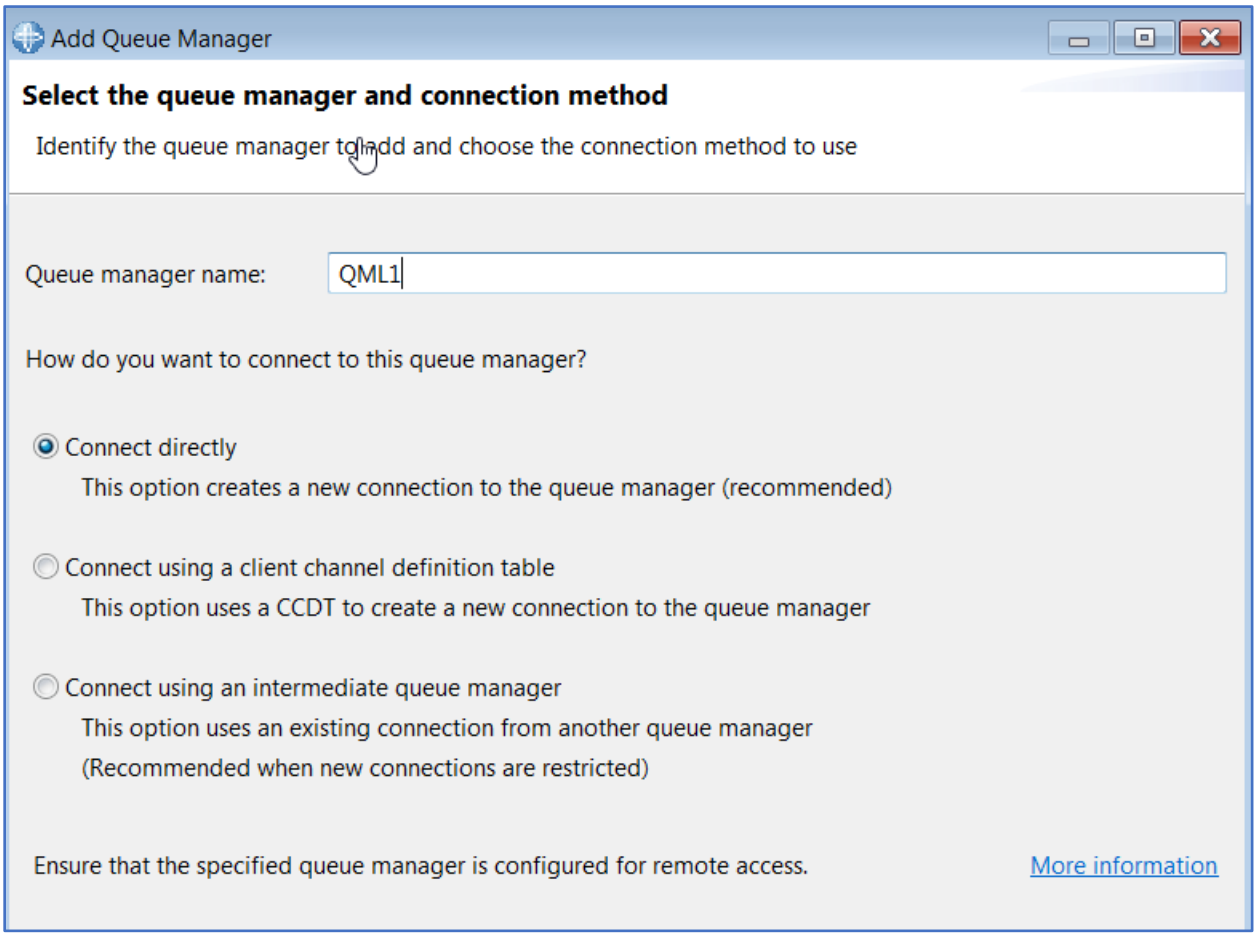
39) Congratulations! You have successfully restored a persistent message to a queue!

Appendix – Adding Queue Managers to the MQ Explorer

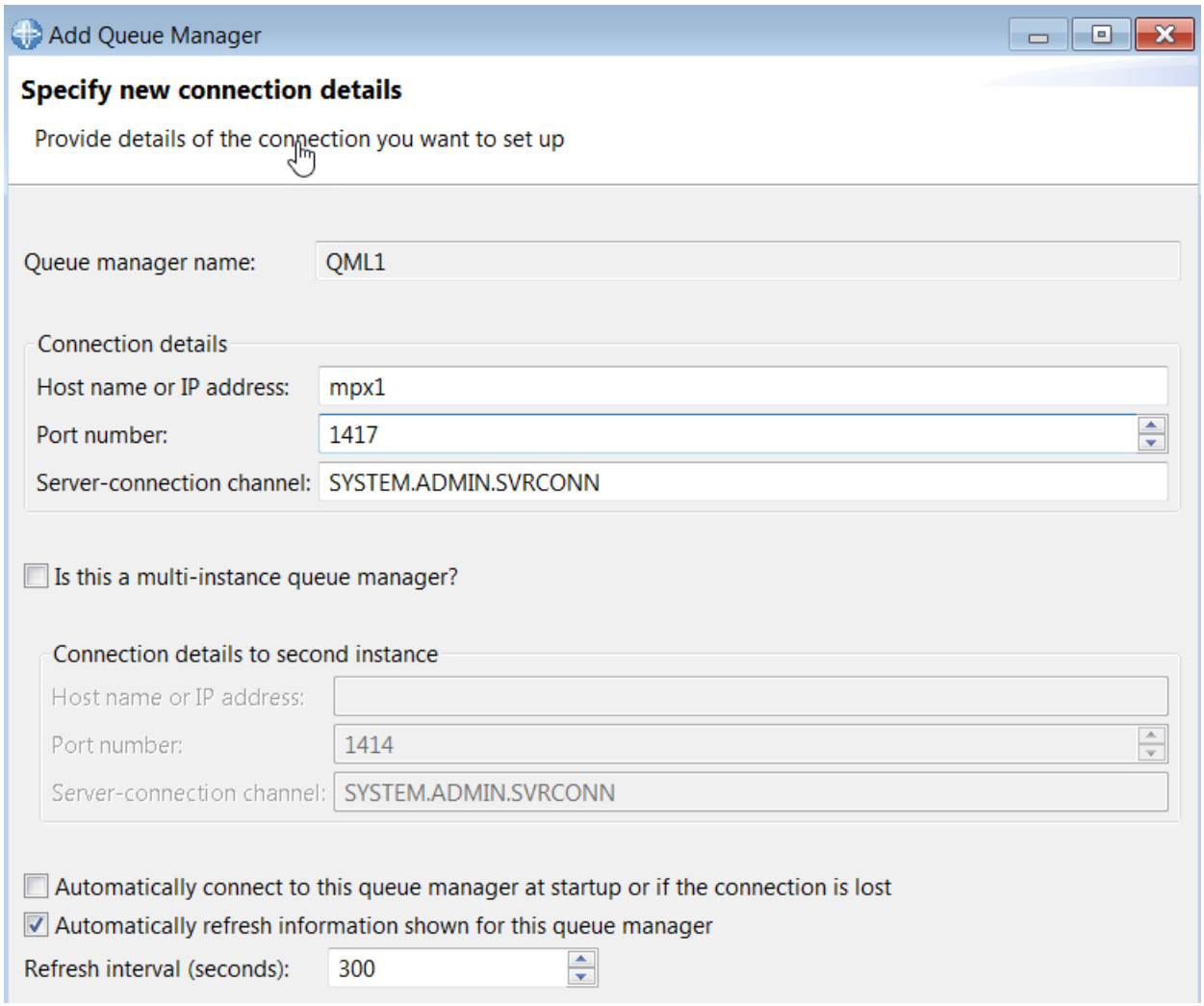
1) Right click on the 'Queue Managers' folder and select 'Add Remote Queue Manager'



2) Enter QML1 (we will add both queue managers used) and select 'Connect directly'. Then click on the 'Next' button (not shown, but it's on the bottom of the panel).



3) Enter 'mpx1' for the Host name or IP address and 1417 as the port number. Then click on the 'Finish' button.



4) Repeat the steps for QML2, using mpx2 as the host name and 1418 as the port.

5) The Queue Manager list should now include both the z/OS queue managers used for this lab.