MQ SMF –Data collection and running MP1B

Table of Contents

Table of Contents

Lab Objective

One objective is to familiarize you with the output of both OEMPUT and the MP1B SMF formatter. A second objective is to give a sample that can be run in your environment to compare costs based on application programming styles.

Lab Information

1) Use the primary queue manager assigned on the worksheet
2) Logon to MPX1 or MPX2 as directed on the worksheet
3) You will be tailoring jobs within your assigned JCL libraries and submitting them

Lab Steps

I. Data Capture – using OEMPUT to compare MQPUT1 and MQPUT in loops.

Tech Tip: OEMPUT is a program included with SupprtPac MP1B and was originally designed to be a flexible framework for testing messaging and transformation (by what was known at the time as Message Broker on z/OS). It works extremely well for compare/contract infrastructure and application scenarios. The WSC recommends creating a series of tests using this tool for regression testing.

For this exercise, you may choose to capture the SMF data yourself or to copy the data from a capture done previously. If you are copying the data, please skip to "Using IFASMFDP to copy the sample data. "

A. Tailoring the Queue Definitions and defining them.

a. In ISPF edit the JCL library for this workshop, called TEAMXX.MQPERF.JCL replacing the TEAMXX with your user ID from the worksheet.
b. Select member DEFQS, this member will define the queues used in this exercise. The member will look something like this:

DO NOT USE THE MQPREF.IMQPF.JCL!!! Use your JCL PDS TEAMXX.MQPERF.JCL, replacing TEAMXX with your ID

```
EDIT       MQPERF.IMQPF.JCL(DEFQS) - 01.00                Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //++TEAMXX++D JOB (????,????),'DEFINES',NOTIFY=????????
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 //*
000012 //DEFQS    EXEC PGM=CSQUTIL,PARM='++QMGR++'
000013 //STEPLIB  DD DISP=SHR,DSN=++MQHLQ++.SCSQANLE
000014 //         DD DISP=SHR,DSN=++MQHLQ++.SCSQAUTH
000015 //         DD DISP=SHR,DSN=++MQHLQ++.SCSQLOAD
000016 //SYSIN    DD *
000017  COMMAND
000018 /*
```

c. Do a change all for each of the '++' variables listed in the JCL. For example, to change the MQ high level qualifier, enter the command

```
c ++MQHLQ++ MQ910 all
```

You MUST remove the '++' characters from the JCL. Using the change command as shown above for all the variables will do that without further modification.

The ++ variables used in this JCL are:
++MQHLQ++
++TEAMXX++
++LPAR++

++QMGR++

d. Save and submit the job, the two commands can be concatenated via a semicolon as shown:

e. Once the job has completed, split the ISPF session by using the 'split' command. On the second session enter the command '=SDSF.ST' to review the output.

f. You may need to set the prefix for your jobs to be displayed, use the 'PREFIX TEAMXX*'command- replacing the XX with your team ID.

g. Use a question mark beside the job to display the output files:

h. Select the SYSPRINT DD name by putting a character in the 'NP' column, and 's' is shown in the example.

i. The example output shown here is for my user ID, ELKINSC.

```
 CSQU000I CSQUTIL IBM MQ for z/OS V9.0.5
 CSQU001I CSQUTIL Queue Manager Utility - 2018-11-07 18:12:52
 COMMAND
 CSQU127I  Executing COMMAND using input from CSQUCMD data set
 CSQU120I  Connecting to QML1
 CSQU121I  Connected to queue manager QML1
 CSQU055I  Target queue manager is QML1
  DEFINE QLOCAL('ELKINSC.TEST.MQPUT1') QSGDISP(QMGR)                    -
         STGCLASS(STGCLS02) REPLACE
CSQN205I   COUNT=       2, RETURN=00000000, REASON=00000000
CSQ9022I QML1 CSQMAQLC ' DEFINE QLOCAL' NORMAL COMPLETION
```

j. Verify that the return code and reason code are 0 for each DEFINE QLOCAL command. Even when the job return code is zero, the CSQUTIL commands may contain errors that need to be corrected before going to the next step.

B. Tailoring the test job and running it.

a. Switch back to the JCL PDS and from the list of members, select the PUT1TEST member. It should look something like this:

```
EDIT       MQPERF.IMQPF.JCL(PUT1TEST) - 01.06              Columns 00001 0007
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //++TEAMXX++P JOB (????,????),'DEFINES',NOTIFY=????????
000002 /*JOBPARM SYSAFF=(++LPAR++)
000003 //*
000004 //* This JCL executes OEMPUT twice. The first 2 minute execution
000005 //* uses MQPUT1 to put all the messages.  The second will open
000006 //* the queue, put the message for the duration, then close the
000007 //* queue.  Each execution puts 25 messages, then gets 25 messages.
000008 //*
000009 //* MAKE THE FOLLOWING CHANGES TO THE JCL:
000010 //*  1) CHANGE ++TEAMXX++ TO YOUR TEAM ID
000011 //*  2) CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
000012 //*  3) CHANGE ++QMGR++ TO THE QMGR ON YOUR WORKSHEET
000013 //*  3) CHANGE ++MQHLQ++ TO THE MQ HIGH LEVEL QUALIFIER
000014 //*
000015 //  SET M=++QMGR++
000016 //  SET Q=++TEAMXX++.TEST.PUT1
000017 //  SET R=++TEAMXX++.TEST.PUT1
000018 //  SET L=1000
```

b. Change all the ++ variables to the values from the worksheet. In this JCL the ++ variables are:
   ++TEAMXX++
   ++LPAR++
   ++QMGR++
   ++MQHLQ++
c. Save and submit the job.
d. The job is in two steps and will run for 4 to 5 minutes before completing. Have a coffee break!!

e. Once the job has completed, move to the SDSF.ST panel to review the output. Select the first 'SYSPRINT' output file, for step PUT01A as shown.

```
 Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF JOB DATA SET DISPLAY - JOB ELKINSCP (JOB06602)     DATA SET DISPLAYED
COMMAND INPUT ===>                                        SCROLL ===> CSR
NP   DDNAME   StepName ProcStep DSID Owner    C Dest              Rec-Cnt Page
     JESMSGLG JES2                2 ELKINSC  S LOCAL                23
     JESJCL   JES2                3 ELKINSC  S LOCAL                50
     JESYSMSG JES2                4 ELKINSC  S LOCAL               119
s_   SYSPRINT PUT01A             103 ELKINSC  S LOCAL                45
     SUMMARY  PUT01A             104 ELKINSC  S LOCAL                 1
     SYSPRINT PUT02A             105 ELKINSC  S LOCAL                45
     SUMMARY  PUT02A             106 ELKINSC  S LOCAL                 1
     SYSOUT   PUT01A             107 ELKINSC  S LOCAL                 1
     SYSOUT   PUT02A             108 ELKINSC  S LOCAL                 1
```

f. The first part of the output should look something like this:

```
Compiled Nov  1 2018 19:44:10.
parm: -mQML3 -cgpc -put1 -qELKINSC.TEST.PUT1 -rELKINSC.TEST.PUT1 -s1000 -n-1 -l2
Test will run for 2 minutes
Message file -FILE: DD:MIN open mode:rb
bytes read from msg file  80
reply size 104857600
OEMPUT about to MQCONN to QMgr QML3.
CPU type 0000012827
Date Time 2018/11/13 17:11:42.
Using MQPUT1
Entering PUT/GET loops (MQGET_Wait=60 seconds)...
Preload the queue with 0 messages...
  Message size        : 1000
  Reply size          : 104857600
  Message persistence : NON-PERSISTENT
  Messages per loop   : 25
  Total messages      : -1
  Syncpoints Get 25,  Put 25, Commit in syncpoint
```

Note the information about the test attributes; including things like the length of time the test will run, the ‘Using MQPUT1’, the number of messages per loop, etc.

g. Page forward (F8 key) and make note of some of the results below (your numbers are very likely to be a bit different).

```
  MQGET replies by    : Any message
Starting loop at 2018-11-13 17:11:42.890201
 Workload manager data
                  Samples %idle %unknown(MQ?) %using CPU %doing I/O %Wait for CP
-------------------------------------------------------
Total Transactions  : 2277800
Elapsed Time        :  120.001 seconds
Application CPU Time:  115.379 seconds  (96.1%)
Transaction Rate    : 18981.443 trans/sec
-------------------------------------------------------
Round trip per msg  :       52 microseconds
Avg App CPU per msg :       50 microseconds
```

Total Transactions _____
Elapsed Time: _____
Application CPU Time: _____
Transaction Rate: _____

h. Return to the list of output DDs and select the second SYSPRINT output, for JOBSTEP PUT02A.
i. You should see that the job ran for 2 minutes, but no indication that MQPUT1 was in use.
j. Page forward (F8).

k. The total transactions and transaction rate is likely to noticeably higher than the previous test.

```
------------------------------------------------------
Total Transactions   : 2829625
Elapsed Time         :  120.001 seconds
Application CPU Time:  115.013 seconds  (95.8%)
Transaction Rate     : 23579.953 trans/sec
------------------------------------------------------
Round trip per msg   :       42 microseconds
Avg App CPU per msg  :       40 microseconds
------------------------------------------------------
```

Total Transactions _____
Elapsed Time: _____
Application CPU Time: _____
Transaction Rate: _____

l. In the sample test run, according to the sample output, the number of transactions was about 25% higher. The average CPU used per message was 10 microseconds lower, or about 20% lower.

C. Extracting the SMF data from the live files.

a. The first step is determining which SMF dataset is in use. From the SDSF panel enter the command

```
SDSF STATUS DISPLAY ALL CLASSES
COMMAND INPUT ===> /display SMF_
```

b. The response will look something like this:

```
RESPONSE=MPX1
 IEE974I 20.16.57 SMF DATA SETS 875
           NAME                VOLSER SIZE(BLKS) %FULL  STATUS
         P-MPXCAT.SMF.MPX1.MANX
                               MPXCAT      3000     0  ALTERNATE
         S-MPXCAT.SMF.MPX1.MANY
                               MPXCAT      3000    59  ACTIVE
         S-MPXCAT.SMF.MPX1.MANZ
                               Q70006     24000     0  ALTERNATE
```

c. The ACTIVE SMF dataset is likely the one that contains the data from this test run. If one of the datasets indicate 'Dump Required' there may be some data needed from that file as well. Check with the instructor to see if you may need to include more than one file.

d. From the list of JCL members, select the SMFDUMP member. It should look like this:

```
//++TEAMXX++S JOB (????,????),'DUMP SMF',NOTIFY=????????
/*JOBPARM SYSAFF=(++LPAR++)
//*
//* THIS IFASMFDP JOB DUMPS THE MQ SMF DATA TO YOUR DATASET
//*
//* MAKE THE FOLLOWING CHANGES TO THE JCL:
//*  1) CHANGE ++TEAMXX++ TO YOUR TEAM ID
//*  2) CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
//*  2) CHANGE ++ACTSMF++ TO THE ACTIVE SMF DATASET FROM THE DISPLAY
//*
//SMFDUMP8 EXEC PGM=IFASMFDP
//DUMPINA  DD   DISP=SHR,DSN=MPXCAT.SMF.MPX1.++ACTSMF++
//DUMPOUT  DD  DSN=++TEAMXX++.TESTPUT1.MQSMF01,
//             DISP=(NEW,CATLG,DELETE),
//             VOL=SER=Q70006,
```

e. Change the ++ variables using the change all command. In this JCL the ++ variables are:
++TEAMXX++
++LPAR++
++ACTSMF++

f. Save and submit the job.

g. Skip to the Post Process the SMF data step (Step II), once the job has completed successfully.

D. Using IFASMFDP to copy the sample data.

a. In ISPF edit the JCL library for this workshop, called TEAMXX.MQPERF.JCL replacing the TEAMXX with your user ID from the worksheet.

b. Select member COPYSSMF, this member will define the queues used in this exercise. The member will look something like this:

```
//++TEAMXX++S JOB (????,????),'DUMP SMF',NOTIFY=????????
/*JOBPARM SYSAFF=(++LPAR++)
//*
//* THIS IFASMFDP JOB DUMPS THE MQ SMF DATA TO YOUR DATASET
//*
//* MAKE THE FOLLOWING CHANGES TO THE JCL:
//*  1) CHANGE ++TEAMXX++ TO YOUR TEAM ID
//*  2) CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
//*
//SMFDUMP8 EXEC PGM=IFASMFDP
//DUMPINA  DD   DISP=SHR,DSN=MQPERF.TESTPUT1.MQSMF01
//DUMPOUT  DD  DSN=++TEAMXX++.TESTPUT1.MQSMF01,
//             DISP=(NEW,CATLG,DELETE),
//             VOL=SER=Q70006,
//             RECFM=VB,BLKSIZE=27998,
//             SPACE=(CYL,(10,10),RLSE),UNIT=SYSDA
//SYSPRINT DD   SYSOUT=*
```

c. Using a change all command, change the two '++' variables to the values on your worksheet.

```
c '++MQHLQ++' 'MQ901' ALL
```

The ++ variables you need to change are:
++TEAMXX++
++LPAR++

d. Save and submit the job, the two commands can be concatenated via a semicolon as shown:

```
Command ===> save;sub
```

e. Once the job has completed, split the ISPF session by using the 'split' command. On the second session enter the command '=SDSF.ST' to review the output.

f. You may need to set the prefix for your jobs to be displayed, use the 'PREFIX TEAMXX*'command- replacing the XX with your team ID.

g. Select the output using the '?' in the NP column

h. Select the SYSPRINT DDNAME as shown:

```
NP  DDNAME   StepName ProcSte
    JESMSGLG JES2
    JESJCL   JES2
    JESYSMSG JES2
S   SYSPRINT SMFDUMP8
```

i. The output should look something like this:

```
IFA010I  SMF DUMP PARAMETERS
IFA010I  NOSIGVALIDATE -- DEFAULT
IFA010I  SIGSTRIP -- DEFAULT
IFA010I  END(2400)  -- DEFAULT
IFA010I  START(0000)  -- DEFAULT
IFA010I  DATE(1900000,2099366)  -- DEFAULT
IFA010I  OUTDD(DUMPOUT,TYPE(115,116))  -- SYSIN
IFA010I  INDD(DUMPINA,OPTIONS(DUMP))  -- SYSIN
IFA020I  DUMPOUT   -- ELKINSC.TESTPUT1.MQSMF01
IFA020I  DUMPINA   -- MQPERF.TESTPUT1.MQSMF01
                                                SUMMARY
```

j. Move the 'Summary' line to the top of the page.

```
                                    SUMMARY ACTIVITY REPORT
START DATE-TIME  11/07/2018-13:50:42                        END DATE-TIME
RECORD       RECORDS         PERCENT      AVG. RECORD   MIN. RECORD   MAX.
  TYPE          READ       OF TOTAL           LENGTH       LENGTH
     2             1          .57 %           18.00           18
     3             1          .57 %           18.00           18
   115            80        45.98 %        3,281.00          280
   116            92        52.87 %        2,814.78          436
TOTAL            174          100 %        2,996.98           18
NUMBER OF RECORDS IN ERROR              0
```

E. Shift to the right (F11), to verify the number of records written:

```
SUMMARY ACTIVITY REPORT
                        END DATE-TIME  11/07/2018-13:55:18
     AVG. RECORD   MIN. RECORD   MAX. RECORD       RECORDS
          LENGTH        LENGTH        LENGTH       WRITTEN
           18.00            18            18             1
           18.00            18            18             1
        3,281.00           280        11,136            80
        2,814.78           436         5,532            92
        2,996.98            18        11,136           174
    0
```

II. Post Process the SMF data

In this section you will use the MP1B SMF formatter, called MQSMF, to process the captured SMF data. In this exercise the primary focus will be on the TASK records, which is a bit out of order for the workshop – but this lab exercises several things to make sure there is the proper connectivity, everyone understands how the jobs work, and gives an overview of all the output files from MQSMF. Some of the output will be covered in greater detail in later exercises.

a. In ISPF edit the JCL library for this workshop, called TEAMXX.MQPERF.JCL replacing the TEAMXX with your user ID from the worksheet.
b. Select member MQSMFV9, this member is the JCL to run the MQSMF formatter. It looks something like this:

```
//++TEAMXX++M JOB (????,????),'DEFINES',NOTIFY=????????
/*JOBPARM SYSAFF=(++LPAR++)
//*
//* THIS TASK post processes the MQ SMF data
//*
//* MAKE THE FOLLOWING CHANGES TO THE JCL:
//*  1) CHANGE ++TEAMXX++ TO YOUR TEAM ID
//*  2) CHANGE ++LPAR++ TO THE LPAR ON YOUR WORKSHEET
//*
//S1 EXEC PGM=MQSMF,REGION=0M
//*************************************************************
//* Processes MQ SMF records.
//*************************************************************
//STEPLIB DD DISP=SHR,DSN=MQPERF.MP1B.V9.LOAD
//SMFIN DD DISP=SHR,DSN=++TEAMXX++.TESTPUT1.MQSMF01
//SYSIN DD *
```

c. Use the change all commands to change the ++ variables to those that apply to your SMF files. The ++ variables you need to change in this JCL are:
   ++TEAMXX++
   ++LPAR++
d. Save and submit the job.
e. Navigate to SDSF.ST (or split the panel as described above and navigate). Select the SMF output using a question mark by the jobname. The output should look like this:

DDNAME	StepName	ProcStep	DSID	Owner	C	Dest
JESMSGL	JES2		2	ELKINSC	S	LOCAL
JESJCL	JES2		3	ELKINSC	S	LOCAL
JESYSMSG	JES2		4	ELKINSC	S	LOCAL
MESSAGE	S1		102	ELKINSC	S	LOCAL
BUFF	S1		103	ELKINSC	S	LOCAL
BUFFCSV	S1		104	ELKINSC	S	LOCAL
DATA	S1		105	ELKINSC	S	LOCAL
CF	S1		106	ELKINSC	S	LOCAL
CFCSV	S1		107	ELKINSC	S	LOCAL
DB2	S1		108	ELKINSC	S	LOCAL
EOJ	S1		109	ELKINSC	S	LOCAL
LOCK	S1		110	ELKINSC	S	LOCAL
LOG	S1		111	ELKINSC	S	LOCAL
LOGCSV	S1		112	ELKINSC	S	LOCAL
MSGM	S1		113	ELKINSC	S	LOCAL
MSGMCSV	S1		114	ELKINSC	S	LOCAL
SMDS	S1		116	ELKINSC	S	LOCAL
TASKSUM	S1		117	ELKINSC	S	LOCAL

f. Page forward looking for the TASK output. When found select it to display. It should look something like this:

```
Task statistics

          8 MPX1,QML1,2018/11/07,13:50:42,VRM:900,
          8 QML1 MOVER Jobname:QML1CHIN Userid:MQUSER
          8 Start time Oct 29 08:45:01 2018 Started in a different time interval
          8 Interval   Nov  7 13:49:17 2018 - Nov  7 13:50:18 2018 : 60.817324 s
          8 Other reqs : Count                 2
          8 Other reqs : Avg elapsed time     11 uS
          8 Other reqs : Avg CPU              11 uS
          8 Other reqs : Total ET              0.000022 Seconds
          8 Other reqs : Total CPU             0.000022 Seconds
          8 Commit count                       0
          8 Commit avg elapsed time            0 uS
          8 Commit avg CPU time                0 uS
          8 Wait for buffer pool latch         0.000000 Seconds
          8 Held BP Latch                      0.000000 Seconds
          8 No queue records
```

Note that it is normal to see tasks, like this channel initiator task, in the output. You will have to search for the OEMPUT tasks.

g. Search for the task via the job name, if you ran the test search for 'TEAMXX' replacing the XX with your team name. If you copied the sample data search for 'ELK' as the tasks were run under the ID ELKINS. YOU MAY HAVE TO SEARCH SEVERAL TIMES! The output will look something like this:

```
31 QML3 Batch Jobname:ELKINP1P Userid:ELKINSC
31 Start time Nov  7 13:50:05 2018 Started this interval
31 Interval   Nov  7 13:50:05 2018 - Nov  7 13:50:51 2018 : 46.040836
31 Other reqs : Count                  1
31 Other reqs : Avg elapsed time       8 uS
31 Other reqs : Avg CPU                8 uS
31 Other reqs : Total ET               0.000008 Seconds
31 Other reqs : Total CPU              0.000008 Seconds
31 > Latch 12, Total wait      13038 uS, Waits          32, Name DMCNMS
31 > Latch 12, Avg wait          407 uS, Max          1407 uS,    DMCNMS
31 > Latch 19, Total wait       1367 uS, Waits          13, Name BMXL3
31 > Latch 19, Avg wait          105 uS, Max           132 uS,    BMXL3
31 > Latch 31, Total wait          0 uS, Waits           1, Name SMCPVT
31 > Latch 31, Avg wait            0 uS, Max             0 uS,    SMCPVT
31 Longest latch wait at 0000000000000000 13038 uS
31 Avg Latch time per UOW              0 uS
31 Commit count                    35348
31 Commit avg elapsed time            30 uS
31 Commit avg CPU time                 6 uS
```

h. Page forward and you should see the start of the queue accounting data for this task:

```
7860 Open name                                    ELKINSC.TEST.PUT1
7860 Queue type:QLocal                            ELKINSC.TEST.PUT1
7860 Queue indexed by NONE                        ELKINSC.TEST.PUT1
7860 First Opened    Nov 13 13:11:18 2018         ELKINSC.TEST.PUT1
7860 Last Closed     Nov 13 13:12:45 2018         ELKINSC.TEST.PUT1
7860 Page set ID                         2        ELKINSC.TEST.PUT1
7860 Buffer pool                         1        ELKINSC.TEST.PUT1
7860 Current opens                       1        ELKINSC.TEST.PUT1
7860 Total requests                3281240        ELKINSC.TEST.PUT1
7860 Open Count                          1        ELKINSC.TEST.PUT1
7860 Open Avg elapsed time              48 uS     ELKINSC.TEST.PUT1
7860 Open Avg CPU time                  48 uS     ELKINSC.TEST.PUT1
7860 Get count                     1640614        ELKINSC.TEST.PUT1
7860 Get avg elapsed time               15 uS     ELKINSC.TEST.PUT1
```

i. Page forward again

```
7860 Get count                     1640614        ELKINSC.TEST.PUT1
7860 Get avg elapsed time               15 uS     ELKINSC.TEST.PUT1
7860 Get avg CPU time                   15 uS     ELKINSC.TEST.PUT1
7860 Get avg suspended time              0 uS     ELKINSC.TEST.PUT1
7860 Get total empty pages           22822        ELKINSC.TEST.PUT1
7860 Get TOQ average                   691 uS     ELKINSC.TEST.PUT1
7860 Get TOQ maximum                  2353 uS     ELKINSC.TEST.PUT1
7860 Get TOQ minimum                   475 uS     ELKINSC.TEST.PUT1
7860 Get valid count               1640614        ELKINSC.TEST.PUT1
7860 Get valid destructive         1640614        ELKINSC.TEST.PUT1
7860 Get size maximum                 1000 bytes  ELKINSC.TEST.PUT1
7860 Get size minimum                 1000 bytes  ELKINSC.TEST.PUT1
7860 Get size average                 1000 bytes  ELKINSC.TEST.PUT1
7860 Get Dest-Next                 1640614        ELKINSC.TEST.PUT1
7860 Get not persistent count      1640614        ELKINSC.TEST.PUT1
```

j. Note that your values will vary, because some of these messages were initially loaded on the queue. Page forward again.

```
7860 Put1 count                    1640625        ELKINSC.TEST.PUT1
7860 Put1 avg elapsed time              29 uS     ELKINSC.TEST.PUT1
7860 Put1 avg CPU time                  29 uS     ELKINSC.TEST.PUT1
7860 Put1 avg suspended time             0 uS     ELKINSC.TEST.PUT1
7860 Put1 num non persistent       1640625        ELKINSC.TEST.PUT1
7860 Put size maximum                 1000 bytes  ELKINSC.TEST.PUT1
7860 Put size minimum                 1000 bytes  ELKINSC.TEST.PUT1
7860 Put size average                 1000 bytes  ELKINSC.TEST.PUT1
7860 Curdepth maximum                   25        ELKINSC.TEST.PUT1
7860 Total Queue elapsed time     74153553 uS     ELKINSC.TEST.PUT1
7860 Total Queue CPU used         72524887 uS     ELKINSC.TEST.PUT1
7860 Grand total CPU time        72.984602 S
7860 Grand Elapsed time          76.297161 S
7860 % total busy                       88     B,ELKINSCP," ",
```

k. Look thru the data, how many instances of task records are there for the PUT1 test? Note there will be 2 or 3.

Tech Tip: There are multiple task records because the MQ SMF STATIME is set to 1 minute, and the OEMPUT task is set to run for 2 minutes. This is an atypically short SMF interval, but it may be set that way when trying to isolate a performance problem. If the interval were set to a more typical 15 to 30 minutes, there may only be one set of task records for the OEMPUT run.

Tech Tip: When task records are written, ether to the MAN datasets or sent to the logger MOST of the values are cleared internally. The exception to this is the grand total CPU and elapsed time for the task. That is accumulated until the task ends.

l. If you locate the last occurrences of the PUT1 queue (make sure you fully qualify the name if you are using data from your test – the command will be F TEAMXX.TEST.MQPUT1 LAST), the final task CPU may look something like this:

```
7893 Put1 count                    637175       ELKINSC.TEST.PUT1
7893 Put1 avg elapsed time             29 uS    ELKINSC.TEST.PUT1
7893 Put1 avg CPU time                 29 uS    ELKINSC.TEST.PUT1
7893 Put1 avg suspended time            0 uS    ELKINSC.TEST.PUT1
7893 Put1 num non persistent       637175       ELKINSC.TEST.PUT1
7893 Put size maximum                1000 bytes ELKINSC.TEST.PUT1
7893 Put size minimum                1000 bytes ELKINSC.TEST.PUT1
7893 Put size average                1000 bytes ELKINSC.TEST.PUT1
7893 Curdepth maximum                  25       ELKINSC.TEST.PUT1
7893 Total Queue elapsed time    28797982 uS    ELKINSC.TEST.PUT1
7893 Total Queue CPU used        28159559 uS    ELKINSC.TEST.PUT1
7893 Grand total CPU time       28.338238 S
7893 Grand Elapsed time         29.630829 S
7893 % total busy                      88    B,ELKINSCP," ",
```

m. Do the same thing for the final occurrences of the PUT queue, and check the CPU and elapsed time. Again the command F TEAMXX.TEST.MQPUT LAST will locate the final instance in the SMF data.

```
8010 Put count                    753375       ELKINSC.TEST.PUT
8010 Put avg elapsed time             19 uS    ELKINSC.TEST.PUT
8010 Put avg CPU time                 19 uS    ELKINSC.TEST.PUT
8010 Put suspended time                0 uS    ELKINSC.TEST.PUT
8010 Put + put1 valid count       753375       ELKINSC.TEST.PUT
8010 Put size maximum               1000 bytes ELKINSC.TEST.PUT
8010 Put size minimum               1000 bytes ELKINSC.TEST.PUT
8010 Put size average               1000 bytes ELKINSC.TEST.PUT
8010 Put num not persistent       753375       ELKINSC.TEST.PUT
8010 Curdepth maximum                 25       ELKINSC.TEST.PUT
8010 Total Queue elapsed time   26598106 uS    ELKINSC.TEST.PUT
8010 Total Queue CPU used       26026155 uS    ELKINSC.TEST.PUT
8010 Grand total CPU time      26.227873 S
8010 Grand Elapsed time        27.554461 S
8010 % total busy                     86    B,ELKINSCP," ",
```

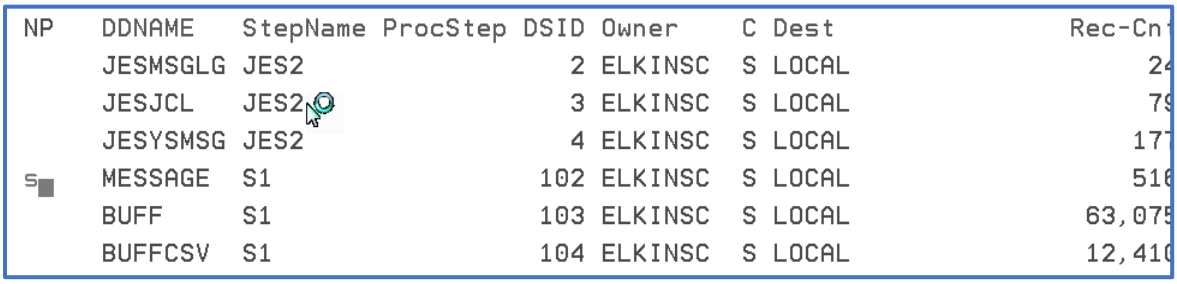
n. The CPU and elapsed times are somewhat different – but the messages counts are substantially different for the PUT1 test and PUT test – which we would expect.

o. If you ran the test yourself, do the CPU counts differ from the OEMPUT totals?

III. Examining the rest of the MQSMF output files

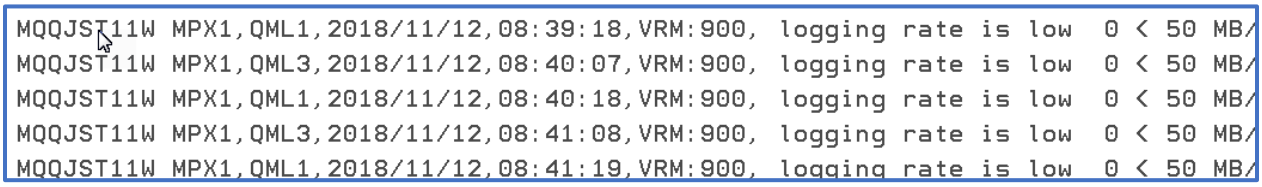
So this lab threw you in the deep end of looking at SMF data first. Now there will be a step back, to look at the higher level information and how the tool can help you see where there may be infrastructure issues.

A. Navigate back to the list of output files from the MQSMF run.



NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cn
	JESMSGLG	JES2		2	ELKINSC	S	LOCAL	2
	JESJCL	JES2		3	ELKINSC	S	LOCAL	7
	JESYSMSG	JES2		4	ELKINSC	S	LOCAL	17
s	MESSAGE	S1		102	ELKINSC	S	LOCAL	51
	BUFF	S1		103	ELKINSC	S	LOCAL	63,07
	BUFFCSV	S1		104	ELKINSC	S	LOCAL	12,41

B. Select the 'MESSAGE' file. You will see messages that are messages that indicate how MQ is running. For example:



```
MQQJST11W MPX1,QML1,2018/11/12,08:39:18,VRM:900, logging rate is low  0 < 50 MB/
MQQJST11W MPX1,QML3,2018/11/12,08:40:07,VRM:900, logging rate is low  0 < 50 MB/
MQQJST11W MPX1,QML1,2018/11/12,08:40:18,VRM:900, logging rate is low  0 < 50 MB/
MQQJST11W MPX1,QML3,2018/11/12,08:41:08,VRM:900, logging rate is low  0 < 50 MB/
MQQJST11W MPX1,QML1,2018/11/12,08:41:19,VRM:900, logging rate is low  0 < 50 MB/
```

C. In this example there may be no messages that are of interest, but there could be. You may see severe messages like MMQQPST01S, which indicate that a bufferpool may have hit short on storage a number of times. My expectations are that when I run this test independently, there are no issues; when a group of people are running the test there may be some buffer impacts shown.

D. Return to the list of output files and select the 'BUFF' file.

```
Buffer statistics
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
  From 2018/11/12,08:38:06.149928 to 2018/11/12,08:39:06.967376, duration    61
= BPool   0, Size    50000,%full now  0, Highest %full  0, Disk reads        0 M
 > 00 Buffs    50000  Low    49930  Now    49930  Getp         2  Getn        0
   00 Rio          0  STW        0  TPW        0  WIO          0  IMW         0
   00 DWT          0  DMC        0  STL        0  STLA         0  SOS         0
   00 Below the bar   PAGECLAS 4KB
= BPool   1, Size    20000,%full now 76, Highest %full 76, Disk reads        0 M
 > 01 Buffs    20000  Low     4615  Now     4615  Getp         7  Getn        0
   01 Rio          0  STW        0  TPW        0  WIO          0  IMW         0
   01 DWT          0  DMC        0  STL        0  STLA         0  SOS         0
   01 Below the bar   PAGECLAS 4KB
= BPool   2, Size    50000,%full now  0, Highest %full  0, Disk reads        0 M
 > 02 Buffs    50000  Low    49988  Now    49988  Getp         1  Getn        0
   02 Rio          0  STW        0  TPW        0  WIO          0  IMW         0
   02 DWT          0  DMC        0  STL        0  STLA         0  SOS         0
   02 Above the bar   PAGECLAS 4KB
```

This report shows the bufferpool use during the interval, paging to the right will display some additional information – but not any of the significant fields. The fields that may be of interest include:

Highest % full

Disk Reads

IMW – Immediate writes to the pagesets done

DWT – Number of times the deferred write threshold was met during the interval (85% full)

DMC – Number of times the synchronous write threshold was met during the interval (95% full)

SOS – Number of times the bufferpool was completely full during the interval

E. The next output file is the BUFFCSV and is a CSV formatted version of the BUFF data. It can be downloaded for a spreadsheet format of the data.

F. DATA – is the report from the data manager

```
Data Manager statistics
MPX1,QML1,2018/11/12,08:39:18,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels         0,  Obj Gets        16
   Locates          9, Stgclass         0,  Enum            14
   Msg Gets        29, Msg Puts         1
   Lock MM          1, Rel MM           2,  Delete MM        0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk        0,        BP          0
MPX1,QML3,2018/11/12,08:40:07,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels        12,  Obj Gets        12
   Locates          0, Stgclass         0,  Enum            12
   Msg Gets         1, Msg Puts         0
   Lock MM          0, Rel MM           0,  Delete MM        0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk        0,        BP          0
MPX1,QML1,2018/11/12,08:40:18,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels         0,  Obj Gets        12
   Locates          0, Stgclass         0,  Enum            12
   Msg Gets         0, Msg Puts         0
   Lock MM          0, Rel MM           0,  Delete MM        0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk        0,        BP          0
```

This data may be interesting to MQ administrators to make sure that there are no unexpected changes to MQ objects. The counts that are interesting are the object creates, the object puts – which are alters to existing objects, and the object deletes. The object gets are when an object is being displayed. Another value that may be interesting is the number of locates, locates are done when an object is being opened or being displayed – which may be prior to it being altered.

Paging thru the data, you may see something like these:

```
MPX1,QML3,2018/11/13,13:13:09,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels       204,  Obj Gets   1640696
   Locates    3281320, Stgclass         0,  Enum            35
   Msg Gets   1640718, Msg Puts   1640727
   Lock MM          0, Rel MM           0,  Delete MM        0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk        0,        BP      22833
```

```
MPX1,QML3,2018/11/13,13:14:10,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels       144,  Obj Gets    637234
   Locates    1274410, Stgclass         0,  Enum            26
   Msg Gets   1279241, Msg Puts   1279234
   Lock MM          0, Rel MM           0,  Delete MM        0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk        0,        BP      17802
```

This was during the PUT1 test, and shows the number of locates, message gets and message puts that were done.

Paging further the intervals where the PUT test was done are also in this data (note

there could be some overlap as this is per interval, not per task.

```
MPX1,QML3,2018/11/13,13:15:10,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels      144,  Obj Gets        58
   Locates         56, Stgclass         0,  Enum           26
   Msg Gets   1434370, Msg Puts   1434359
   Lock MM          0, Rel MM           0,  Delete MM      0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk      0,       BP      19960
```

```
MPX1,QML3,2018/11/13,13:16:11,VRM:900,
   Obj Cre          0, Obj Puts         0,  Obj Dels      156,  Obj Gets        53
   Locates         52, Stgclass         0,  Enum           25
   Msg Gets    753461, Msg Puts    753453
   Lock MM          0, Rel MM           0,  Delete MM      0
   Read Ahead:IO    0,:Buffer           0,  Gets:disk      0,       BP      10489
```

Note the substantial difference between the number of locates done, because the MQPUT1 includes an MQOPEN the locate is done for every instance of MQPUT1.

G. The CF output file contains information about the use of coupling facility structures. Because there has been no shared queue use in this test, the data is very limited in this sample report.

```
cf     statistics
MPX1,QML1,2018/11/12,08:39:18,VRM:900,
  CSQSYSAPPL   , Structure-fulls      0
    Multiple     14, avg et in uS    32, Retries       0
MPX1,QML3,2018/11/12,08:40:07,VRM:900,
  CSQSYSAPPL   , Structure-fulls      0
    Multiple     13, avg et in uS    31, Retries       0
```

There will be additional labs that look into shared queue use later, so we will come back to this report, it's CSV counterpart, along with the DB2, and SMDS reports.

H. The EOJ report is the number of tasks that ended normally and abnormally. It may be used by IBM service.

```
Subsystem statistics
MVS  QMGR Date       Time     VRM          Jobs EOT      Jobs EOM
MPX1,QML1,2018/11/12,08:39:18,VRM:900,           2             0
MPX1,QML1,2018/11/12,08:40:18,VRM:900,           2             0
MPX1,QML1,2018/11/12,08:41:19,VRM:900,           2             0
MPX1,QML1,2018/11/12,08:42:20,VRM:900,           2             0
MPX1,QML1,2018/11/12,08:43:21,VRM:900,           2             0
MPX1,QML1,2018/11/12,08:44:22,VRM:900,           2             0
```

I. The LOCK report is documented as only really being useful to IBM. These locks are used serialize access to resources when necessary.

```
Lock statistics
                                                  Gets Already Held      Releases
MPX1,QML1,2018/11/12,08:39:18,VRM:900,             152            0            61
MPX1,QML3,2018/11/12,08:40:07,VRM:900,           10283            0         10258
MPX1,QML3,2018/11/12,08:41:08,VRM:900,              24            0             0
MPX1,QML3,2018/11/12,08:42:09,VRM:900,              24            0             0
MPX1,QML1,2018/11/12,08:42:20,VRM:900,               1            0             0
```

J. The LOG and LOGCSV reports produced from this data does not contain any interesting information because we were not testing persistent messages. This will be revisited in a later lab.

```
PX1,QML1,2018/11/12,08:39:18,VRM:900,
 From 2018/11/12,08:38:08.821259 to 2018/11/12,08:39:18.054993, duration    69 seconds.
  Wait for buffers(should be 0):             0 out of         0, 0%
  Total Number of pages written:             4
  Number of pages written/sec:               0
  Amount of data written/sec:                0 MB/Sec                      MPX1,QML1,2018/11/12,08:39:18,VRM:900,
  Total Number of write requests:            4
  Number of write requests/sec:              0
  Pages written per I/O:                     1
  Total number of read requests:             0
  Write_Wait         0, Write_Nowait        26, Write_Force          0, WTB         0
  Read_Stor          0, Read_Active          0, Read_Archive         0, TVC         0
  BSDS_Reqs          0, CIs_Created          0, BFWR                 4, ALR         0
  ALW                0, CIs_Offload          0, LLCheckpoints        0
  Read delayed       0, Tape Lookahead       0, Lookahead Mount      0
  Write_Susp         4, Write_Reqs           4, CI_Writes            4
  Write_Serl         0, Write_Thrsh          0, Buff_Pagein          0
  _____,__ write requests,        CIs, Average I/O , After I/O   , pages/IO
                                       time in uSec, time in uSec,
  Log 1, 1 page          4,          4,         269,             1,        1 MPX1,QML1,2018/11/12,08:39:18,VRM:900,
```

K. The MSGM and MSGMCSV, or message manager, reports do contain some interesting data for this lab. Page down until you find a significant number of MQPUT1s for your primary queue manager. Note that the number you see may be quite a bit higher because multiple people are testing, but it will look something like this:

```
MPX1,QML3,2018/11/13,13:13:09,VRM:900,
  From 2018/11/13,13:11:42.889304 to 2018/11/13,13:13:09.322980, duration    86 seconds.
   MQOPENs         35,   MQCLOSEs         34,   MQGETs   1640706,   MQPUTs       102
   MQPUT1s   1640625,   MQINQs           53,   MQSETs         0,   C ALL H        0
   MQSUBs          0,   MQSUBRQs         0,   MQCBs          34
   MQCTLs        242,   MQSTATs          0,   Publish        0
MPX1,QML3,2018/11/13,13:13:09,VRM:900,  Get rate 19077/sec  Put+put1 rate 19078/sec
```

This report has the total number of MQ API requests that have been issued during the interval. In the above example there were 1,640,625 MQPUT1 requests, 102 MQPUT requests, and 1,640,706 MQGET requests.

Tech Tip: These are counts of the number of MQ operations, not a count of the number of successful operations.

L. The TASKSUM report (sorry for the font size) gives area that may need additional evaluation from the TASK report. In this example there were some extended latches (locks) and some long DB2 requests – we will work with latching later.

```
Record#    Count    Value Message
     7860        3    29473 MQTASK15S longest latch wait address 0000000000000000
     7919        2     9691 MQTASK15W longest latch wait address 0000000000000000
     2538      256     1274 MQTASK38W Task had long (1274 uSec) DB2 request > user specified value(100 uSec) M,QML1CHIN," ",
     6686      255     1325 MQTASK38W Task had long (1325 uSec) DB2 request > user specified value(100 uSec) M,QML3CHIN," ",
```

M. The TASK, TASKET, and TASKCSV will be covered in a later lab.

N. The TOPIC report contains data about pub/sub. As there was no pub/sub work during this test, the report looks as shown.

```
Topic statistics
MPX1,QML1,2018/11/12,08:39:18,VRM:900,  no data found
MPX1,QML3,2018/11/12,08:40:07,VRM:900,  no data found
MPX1,QML1,2018/11/12,08:40:18,VRM:900,  no data found
MPX1,QML3,2018/11/12,08:41:08,VRM:900,  no data found
MPX1,QML1,2018/11/12,08:41:19,VRM:900,  no data found
```

O. The STG report is only useful to IBM, but looks as follows:

```
MPX1,QML3,2018/11/13,13:13:09,VRM:900,
   Fixed pools   : Created          0,  Deallocated          0
   Fixed segments: Freed            0,  Expanded             0,  Contracted          0
   Varbl pools   : Created         34,  Deallocated         34
   Varbl segments: Freed          136,  Expanded           136,  Contracted          0
                   Getmains        19,  Freemains           19,  Non-zero RCs        0
                   SOS              0,  Contractions         0,  Abends              0
```

P. LOGBUSY is a summary of the LOG report in a CSV format.

Q. The CHINIT statistics contain the information about the numbers of channels in use during the interval. The CHINCSV is the equivalent data in a CSV format.

```
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
From 2018/11/12,08:38:06.152807 to 2018/11/12,08:39:06.970033 duration 60.817225 seconds
QSG............QSGM
Peak number used of current channels...........   2
Peak number used of active channels ...........   1
MAXCHL. Max allowed current channels........... 200
ACTCHL. Max allowed active channels............ 200
TCPCHL. Max allowed TCP/IP channels............ 200
LU62CHL. Max allowed LU62 channels............. 200
Storage used by Chinit.........................  14MB
```

R. The DISP report includes the channel initiator dispatch task use information. The dispatcher tasks communicate between MQ and the networks. This data can be critical to tuning channel initiators doing a lot of interaction with clients and other queue managers. Added in MQ V8 it has helped provide good information to determine when new dispatcher tasks might be required. The DISPCSV report is the CSV rendering of this data.

```
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
From 2018/11/12,08:38:06.152807 to 2018/11/12,08:39:06.970033 duration 60.817225 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
    ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,DISP,      18,   0.0,     0.000226,  0.0,        13,       10
   1,DISP,       4,   0.0,     0.000068,  0.0,        17,       14
   2,DISP,       0,   0.0,     0.000000,  0.0,         0,        0
   3,DISP,       0,   0.0,     0.000000,  0.0,         0,        0
   4,DISP,       0,   0.0,     0.000000,  0.0,         0,        0
Summ,DISP,      22,   0.0,     0.000293,  0.0,        13,       11
   0,DISP, number of channels on this TCB,    0
   1,DISP, number of channels on this TCB,    1
   2,DISP, number of channels on this TCB,    0
   3,DISP, number of channels on this TCB,    0
   4,DISP, number of channels on this TCB,    0
Summ,DISP, number of channels on all TCBs,    1
```

S. ADAP is the channel initiator adapter task use report. The adapter tasks provide the link from the channel initiator address space to the queue manager. Like the dispatcher information, this was added in MQ V8, and has proven really helpful in tuning high-volume environments. ADAPCSV is the CSV form of this data.

```
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
From 2018/11/12,08:38:06.152807 to 2018/11/12,08:39:06.970033 duration 60.817225 seconds
Task,Type,Requests,Busy %,      CPU used,  CPU %,"avg CPU","avg ET"
    ,    ,        ,      ,       Seconds,       , uSeconds,uSeconds
   0,ADAP,       11,    0.0,      0.000277,   0.0,         25,       23
   1,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
   2,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
   3,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
   4,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
   5,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
   6,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
   7,ADAP,        0,    0.0,      0.000000,   0.0,          0,        0
Summ,ADAP,       11,    0.0,      0.000277,   0.0,         25,       23
```

T. The DNS report is for the CHIN task to contact the domain name server for resolution. Not in use in this test environment, the data is still reported as shown. The critical piece of information is the duration of the longest request. DNSCSV is the CSV form of the same data.

```
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
From 2018/11/12,08:38:06.152807 to 2018/11/12,08:39:06.970033 duration 60.817225 seconds
Task,Type,Requests,Busy %,      CPU used,  CPU %,"avg CPU","avg ET",longest ,date      ,time
    ,    ,        ,      ,       Seconds,       , uSeconds,uSeconds,uSeconds,          ,
   0,DNS ,        0,    0.0,      0.000000,   0.0,          0,        0
Summ,DNS ,        0,    0.0,      0.000000,   0.0,          0,        0
```

U. The SSL report is like the other channel initiator task reports, this reports on how busy the task(s) is(are) and like DNS give the duration of the longest request during the interval. SSLCSV is the CSV form of the data.

```
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
From 2018/11/12,08:38:06.152807 to 2018/11/12,08:39:06.970033 duration 60.817225 seconds
Task,Type,Requests,Busy %,      CPU used,  CPU %,"avg CPU","avg ET",longest ,date      ,time
    ,    ,        ,      ,       Seconds,       , uSeconds,uSeconds,uSeconds,          ,
```

V. The DCHS, and the related reports DCHCSV and DCHSSUM, are analogous to the TASK reports for channels. These records are for all types of channels, the information will be different based on the channel type.

DCHS:

```
Jobname: MPX1,QML1,2018/11/13,13:04:13,VRM:900,Last or only record
SMF interval start  local time   2018/11/13,13:03:12
SMF interval end    local time   2018/11/13,13:04:13
SMF interval start  GMT          2018/11/13,17:03:12
SMF interval end    GMT          2018/11/13,17:04:13
SMF interval duration        60.817498 seconds
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Connection name               192.168.215.141
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Channel disp                  PRIVATE
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Channel type                  SVRCONN
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Channel status                RUNNING
00000000 : D539EB2F 00000000                    N. .....          .9 /....
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Remote qmgr/app               MQ Explorer 9.1.0
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Channel started date & time   2018/11/13,16:36:09
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Channel status collect time   2018/11/13,17:04:13
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Active for                                 61 seconds
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Last MQI request time         2018/11/13,17:03:19
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Dispatcher number                          1
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Number of MQI requests                     5
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Number of persistent messages              0
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Buffers sent                               5
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Buffers received                           4
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Current shared connections                 1
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Message data                               0     0  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Persistent message data                    0     0  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Non persistent message data                0     0  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Total bytes sent                       4,796  4796  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Total bytes received                     888   888  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Bytes received/message                   177   177  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Bytes sent/message                       959   959  B
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Bytes received/second                     14    14  B/sec
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Bytes sent/second                         78    78  B/sec
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Compression rate                           0
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Exit time average                          0 uSec
SYSTEM.ADMIN.SVRCONN 192.168.215.141 DNS resolution time                        0 uSec
SYSTEM.ADMIN.SVRCONN 192.168.215.141 CN from SSLCERT
SYSTEM.ADMIN.SVRCONN 192.168.215.141 Serial number                              00000000 00000000x
```

DCHSSUM:

```
MVS,MQ,date,time,VRM,channelType,count,Persistent,NonPersistent,'P/Sec','NP/Sec'
MPX1,QML1,2018/11/13,13:04:13,VRM:900,SVRCONN,12,0,0,0,0
MPX1,QML1,2018/11/13,13:04:13,VRM:900,CLUSRCVR,2,0,0,0,0
MPX1,QML1,2018/11/13,13:04:13,VRM:900,CLUSSDR,2,0,0,0,0
MPX1,QML1,2018/11/13,13:04:13,VRM:900,total,16,0,0,0,0
MPX1,QML1,2018/11/13,13:05:13,VRM:900,SVRCONN,12,0,0,0,0
```

DCHSCSV – Note that at the time of the writing, there is an open question to the development lab about why the SVRCONNs are not represented in the data:

```
MVS,MQ,date,time,VRM,channelType,ChannelName,Address,BSZ,ABSZ,Bytes/sec
MPX1,QML1,2018/11/13,13:04:13,VRM:900,CLUSSDR,"TO.QML4          "," 192.168.17.253 ",50,0.0,0.0
MPX1,QML1,2018/11/13,13:04:13,VRM:900,CLUSRCVR,"TO.QML1          "," 192.168.17.253 ",50,0.0,0.0
MPX1,QML1,2018/11/13,13:04:13,VRM:900,CLUSSDR,"TO.QML2          "," 192.168.17.253 ",50,0.0,0.0
MPX1,QML1,2018/11/13,13:04:13,VRM:900,CLUSRCVR,"TO.QML1          "," 192.168.17.253 ",50,0.0,0.0
MPX1,QML1,2018/11/13,13:05:13,VRM:900,CLUSSDR,"TO.QML4          "," 192.168.17.253 ",50,0.0,0.0
```

W. The SMDS reports, SMDSSCSV, SMDSBCSV, and SMDSACSV will be covered in a later lab.

X. The GETMAIN report contains no data and is not currently described.

Y. The PUTSCSV and PUTSCSVS summarize the MQPUT and MQPUT1 activity. The PUTSCSV is a per task record summary of put activity by queue, the PUTSCSVS is the summary by the queue name.

```
MVS ,QM,SMF_Date,SMF_Time,TranT,Tran1,Tran2,Queue,Puts,Put1s,TotBytes,MaxMsgSz,MinMsgSz
MPX1,QML3,2018/11/13,13:13:09,B,ELKINSCP," ",ELKINSC.TEST.PUT1                    ,0,1640625,1640625000,1000,1000,
MPX1,QML3,2018/11/13,13:13:42,B,ELKINSCP," ",ELKINSC.TEST.PUT1                    ,0,637175,637175000,1000,1000,
MPX1,QML3,2018/11/13,13:14:10,B,ELKINSCP," ",ELKINSC.TEST.PUT                     ,641975,0,641975000,1000,1000,
MPX1,QML3,2018/11/13,13:15:10,B,ELKINSCP," ",ELKINSC.TEST.PUT                     ,1434275,0,1434275000,1000,1000,
MPX1,QML3,2018/11/13,13:15:42,B,ELKINSCP," ",ELKINSC.TEST.PUT                     ,753375,0,753375000,1000,1000,
```

```
Queue,Puts,Put1s,TotBytes,MaxMsgSz,MinMsgSz
ELKINSC.TEST.PUT                                  ,2829625,0,2829625000,1000,1000,
ELKINSC.TEST.PUT1                                 ,0,2277800,2277800000,1000,1000,
```

Z. The GETSCSV and GETSCSVS summarize the MQGET activity. The GETSCSV is the per task record set information and the GETSCSVS is the summary by the queue name.

```
MVS ,QM,SMF_Date,SMF_Time,TranT,Tran1,Tran2,Queue,TotalGets,ValidGets,TotBytes,MaxMsgSz,MinMsgSz,MaxLat(us),MinLat(us),AvgLat(us)
MPX1,QML3,2018/11/13,13:13:09,B,ELKINSCP," ",ELKINSC.TEST.PUT1                    ,1640614,1640614,1640614000,1000,1000,2
MPX1,QML3,2018/11/13,13:13:42,B,ELKINSCP," ",ELKINSC.TEST.PUT1                    ,637186,637186,637186000,1000,1000,1549
MPX1,QML3,2018/11/13,13:14:10,B,ELKINSCP," ",ELKINSC.TEST.PUT                     ,641969,641969,641969000,1000,1000,1660
MPX1,QML3,2018/11/13,13:15:10,B,ELKINSCP," ",ELKINSC.TEST.PUT                     ,1434281,1434281,1434281000,1000,1000,7
MPX1,QML3,2018/11/13,13:15:42,B,ELKINSCP," ",ELKINSC.TEST.PUT                     ,753375,753375,753375000,1000,1000,1518
```

```
Queue,TotalGets,ValidGets,TotBytes,MaxMsgSz,MinMsgSz,MaxLat(us),MinLat(us),AvgLat(us)
ELKINSC.TEST.PUT                                  ,2829625,2829625,2829625000,1000,1000,7203,470,2186,
ELKINSC.TEST.PUT1                                 ,2277800,2277800,2277800000,1000,1000,2353,474,8721,
```

AA. The QSUML and QSUMS reports summarize the activity on local private queues and shared queues. As this test did not include any share queue activity, the QSUMS report will not include any activity.

```
Queue tree
Date,Time,Qmgr,Queue,          ,Count,PS,BP,"Put MB","Get MB",!,ValidPut,ValidGet,getpsn,MaxQDepth,TotalGets,
2018/11/13,13:00:00,QML3,ELKINSC.TEST.PUT                                ,   3, 1,   2,2.7e+03,2.7e+03,!,2.82963e+06,2.82963e+06,0,25
2018/11/13,13:00:00,QML3,ELKINSC.TEST.PUT1                               ,   2, 1,   2,2.2e+03,2.2e+03,!,2.2778e+06,2.2778e+06,0,25,2
```

```
Queue tree
Date,Time,Qmgr,,Queue,          Count,Structure    ,"Put MB","Get MB",!,ValidPut,ValidGet,MaxQDepth,TotalGets,
********************************* BOTTOM OF DATA ********************************************************************************
```

BB. The STGSUM is a summary report of storage utilization in the queue managers. Some of the data is only of interest to IBM. For a complete description of the fields of interest, please see the MP1B documentation. At the time of writing, the JES log messages CSQY220I are much clearer on storage use than this report.

```
MPX1,QML3,2018/11/12,08:40:07,VRM:900,
  From 2018/11/12,08:38:08.821259 to 2018/11/12,08:39:18.054993, duration    60 seconds.
Reason:  Statistics interval
>16MB Used   598 MB Free  1134 MB %used  34 delta    0 MB
Real Used                          610 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Pool global fixed,        364544,     0 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Pool global variable,      81920,     0 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Pool local fixed    ,    1261568,     1 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Pool local variable,     1290240,     1 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Total stack storage      5021696,     4 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Getmained          ,     7983273,     7 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Avail 64 storage   ,  1663041536,     1 GB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Cushion 64 bit     ,   321912832,     0 GB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, <16MB Allocated user,   00011000,     0 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, <16MB Allocated system  00064000      0 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, <16MB Max size user  ,  008fa000,     8 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, <16MB Low start     ,   00006000,
MPX1,QML3,2018/11/12,08:40:07,VRM:900, <16MB current user high,00020000,
MPX1,QML3,2018/11/12,08:40:07,VRM:900, <16MB not used          ,          8 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Region >16MB size       ,       1733 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, >16MB Allocated user  , 0109e000,   16 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, >16MB Allocated system, 24585000,   581 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, >16MB Max limit        ,6c500000,  1733 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, >16MB Start of region  ,13b00000,
MPX1,QML3,2018/11/12,08:40:07,VRM:900, >16MB Current High     ,14ba4000,
MPX1,QML3,2018/11/12,08:40:07,VRM:900, >16MB not used                  ,  1134 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, QM available                    ,  1134 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Reserved for MVS       ,   1048576,   1 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Critical level         ,   1048576,   1 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, SOS Cushion            , 104857600, 100 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, ASID mem limit                  ,     3 GB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Object storage                     1858 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Object storage hwater              1114 MB
MPX1,QML3,2018/11/12,08:40:07,VRM:900, Number of objects          ,       429
Why limited:Set in the JCL
```

CC. The PSET and PSETCSV reports were introduced with MQ V8, when statistics were added for pageset utilization. Note that there was little pageset activity during this test – certainly no expansion.

```
MPX1,QML3,2018/11/12,08:39:06,VRM:900,
  From 2018/11/12,08:38:06.149928 to 2018/11/12,08:39:06.967376, duration    61 seconds.
PS00 BP   0, Pages      5038, Size   19 MB, free    98.4%, used     1.6%, P  1%, NP  0%, #full 0,
PS00 No I/O activity
PS01 BP   0, Pages      5038, Size   19 MB, free    99.1%, used     0.9%, P  0%, NP  0%, #full 0,
PS01 No I/O activity
PS02 BP   1, Pages    111232, Size  434 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS02 No I/O activity
PS03 BP   2, Pages      1078, Size 4312 KB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS03 No I/O activity
PS04 BP   3, Pages     86393, Size  337 MB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS04 No I/O activity
PS10 BP  10, Pages      1078, Size 4312 KB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS10 No I/O activity
PS11 BP  11, Pages      1078, Size 4312 KB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS11 No I/O activity
PS12 BP  12, Pages      1078, Size 4312 KB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS12 No I/O activity
PS13 BP  13, Pages      1078, Size 4312 KB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
PS13 No I/O activity
PS14 BP  14, Pages      1078, Size 4312 KB, free   100.0%, used     0.0%, P  0%, NP  0%, #full 0,
```

```
z/OS,QM,Date,Time,StatDur,PS,BP,pages,MB,%free,%used,%P,%NP,#full
MPX1,QML3,2018/11/12,08:39:06,61,00,00,5038,19.7,98.4,1.6, 1, 0,0,
MPX1,QML3,2018/11/12,08:39:06,61,01,00,5038,19.7,99.1,0.9, 0, 0,0,
MPX1,QML3,2018/11/12,08:39:06,61,02,01,111232,434.5,100.0,0.0, 0, 0,0,
MPX1,QML3,2018/11/12,08:39:06,61,03,02,1078,4.2,100.0,0.0, 0, 0,0,
MPX1,QML3,2018/11/12,08:39:06,61,04,03,86393,337.5,100.0,0.0, 0, 0,0,
MPX1,QML3,2018/11/12,08:39:06,61,10,10,1078,4.2,100.0,0.0, 0, 0,0,
```

DD. The QMAC report is for the class 1 accounting records. At the time of this writing, that report was not formatted correctly. As this report is seldom used, the class 1 data has not been updated since V5.3.1, this data is often ignored. The problem has been reported and shall be addressed at some point.

```
MPX1,QML3,2018/11/12,08:46:22,VRM:900, QMAC: CPU 00823460     0-99       100-999     1000-9999        >9999
         MQPUT      3891386580      3555917824      4193316864           308
         MQGET       8388609            84      11534337           260
```

EE. The PSIDQIO report is a summary of any I/O to pagesets based on the queue name. For this test the report did not contain any data.

```
********************************** TOP OF DATA ***********************************
                     QMGR, BP, PS,           Count,   Total_time,Avg_time,    Rate,Queue
Last entries
******************************** BOTTOM OF DATA **********************************
```

FF. QALL is a summary of activity by queue. What is shown below is the activity for the PUT test. Which totals would you expect to be different for the PUT1 test?

```
Queue data summarised by queue
          0 Open name                                ELKINSC.TEST.PUT
          0 Queue type:QLocal                        ELKINSC.TEST.PUT
          0 Queue indexed by NONE                    ELKINSC.TEST.PUT
          0 First Opened     Nov 13 13:13:18 2018    ELKINSC.TEST.PUT
          0 Last Closed      Nov 13 13:15:18 2018    ELKINSC.TEST.PUT
          0 Page set ID                         2    ELKINSC.TEST.PUT
          0 Buffer pool                         1    ELKINSC.TEST.PUT
          0 Current opens                       4    ELKINSC.TEST.PUT
          0 Total requests                 11095705  ELKINSC.TEST.PUT
          0 Open Count                          2    ELKINSC.TEST.PUT
          0 Open Avg elapsed time              13 uS ELKINSC.TEST.PUT
          0 Open Avg CPU time                  13 uS ELKINSC.TEST.PUT
          0 Close count                         2    ELKINSC.TEST.PUT
          0 Close avg elapsed time              6 uS ELKINSC.TEST.PUT
          0 Close avg CPU time                  6 uS ELKINSC.TEST.PUT
          0 Get count                     2829625    ELKINSC.TEST.PUT
          0 Get avg elapsed time               15 uS ELKINSC.TEST.PUT
          0 Get avg CPU time                   15 uS ELKINSC.TEST.PUT
          0 Get avg suspended time              0 uS ELKINSC.TEST.PUT
          0 Get total empty pages           39354    ELKINSC.TEST.PUT
          0 Get TOQ average                   566 uS ELKINSC.TEST.PUT
          0 Get TOQ maximum                  7203 uS ELKINSC.TEST.PUT
          0 Get TOQ minimum                   470 uS ELKINSC.TEST.PUT
          0 Get valid count               2829625    ELKINSC.TEST.PUT
          0 Get valid destructive         2829625    ELKINSC.TEST.PUT
          0 Get size maximum                1000 bytes ELKINSC.TEST.PUT
          0 Get size minimum                1000 bytes ELKINSC.TEST.PUT
          0 Get size average                1000 bytes ELKINSC.TEST.PUT
          0 Get Dest-Next                 2829625    ELKINSC.TEST.PUT
          0 Get not persistent count      2829625    ELKINSC.TEST.PUT
          0 Put count                     2829625    ELKINSC.TEST.PUT
          0 Put avg elapsed time               19 uS ELKINSC.TEST.PUT
          0 Put avg CPU time                   19 uS ELKINSC.TEST.PUT
          0 Put suspended time                  0 uS ELKINSC.TEST.PUT
          0 Put + put1 valid count        2829625    ELKINSC.TEST.PUT
          0 Put size maximum                1000 bytes ELKINSC.TEST.PUT
          0 Put size minimum                1000 bytes ELKINSC.TEST.PUT
          0 Put size average                1000 bytes ELKINSC.TEST.PUT
          0 Put num not persistent        2829625    ELKINSC.TEST.PUT
          0 inq count                           1    ELKINSC.TEST.PUT
          0 inq avg elapsed time               13 uS ELKINSC.TEST.PUT
          0 inq avg CPU time                   13 uS ELKINSC.TEST.PUT
          0 Curdepth maximum                   25    ELKINSC.TEST.PUT
          0 Total Queue elapsed time     99802108 uS ELKINSC.TEST.PUT
          0 Total Queue CPU used         97649468 uS ELKINSC.TEST.PUT
```

GG. The SYSPRINT dataset contains information about the values used to drive some notifications and, more importantly, the types and subtypes of records processed. Paging to the bottom of the report (entering the command 'BOT' from the command line will take you to that information.

```
Summary of MQ SMF records and subtypes found
============================================
SMF type 115 subtype   1, record count     516 System statistics(1)
SMF type 115 subtype   2, record count     516 System statistics(2)
SMF type 115 subtype   5, record count     516 Storage statistics
SMF type 115 subtype   6, record count     516 Storage detail statistics
SMF type 115 subtype   7, record count     516 Storage summary statistics
SMF type 115 subtype 201, record count     517 Page set statistics
SMF type 115 subtype 215, record count     517 Buffer manager extension
SMF type 115 subtype 231, record count     516 Chinit statistics
SMF type 116 subtype   0, record count      40 Accounting class(1)
SMF type 116 subtype   1, record count    4104 Accounting class(3)
SMF type 116 subtype  10, record count      17 Channel accounting data
```

This information will tell you what kinds of data is being captured for processing.

HH. The CMESSAGE output is the messages from the channel statistics collection:

```
MQCHIN002I MPX1,QML3,2018/11/12,08:39:06,VRM:900, High water mark of current channels (2) 1% of MAXCHL(200)
MQCHIN013I MPX1,QML3,2018/11/12,08:39:06,VRM:900, High water mark of active channels (1) 0% of ACTCHL(200)
MQCHIN002I MPX1,QML1,2018/11/12,08:39:18,VRM:900, High water mark of current channels (1) 0% of MAXCHL(200)
MQCHIN013I MPX1,QML1,2018/11/12,08:39:18,VRM:900, High water mark of active channels (1) 0% of ACTCHL(200)
MQCHIN002I MPX1,QML3,2018/11/12,08:40:07,VRM:900, High water mark of current channels (2) 1% of MAXCHL(200)
MQCHIN013I MPX1,QML3,2018/11/12,08:40:07,VRM:900, High water mark of active channels (1) 0% of ACTCHL(200)
MQCHIN002I MPX1,QML1,2018/11/12,08:40:18,VRM:900, High water mark of current channels (1) 0% of MAXCHL(200)
MQCHIN013I MPX1,QML1,2018/11/12,08:40:18,VRM:900, High water mark of active channels (0) 0% of ACTCHL(200)
MQCHIN002I MPX1,QML3,2018/11/12,08:41:08,VRM:900, High water mark of current channels (2) 1% of MAXCHL(200)
MQCHIN013I MPX1,QML3,2018/11/12,08:41:08,VRM:900, High water mark of active channels (1) 0% of ACTCHL(200)
```

Additional Information

The MP1B SupportPac can currently be found here:

http://www-01.ibm.com/support/docview.wss?uid=swg24005907

I expect that it will be moved to the IBM Messaging GitHub page when it is next published.

The IBM messaging performance GitHub page may be found here:

https://github.com/ibm-messaging/mqperf

The IBM messaging GitHub page may be found here:

https://github.com/ibm-messaging