



## **Looking at SMF data for problem determination**

Audience level: Some knowledge of MQ or z/OS

Skillset: MQ Administration, z/OS systems programming

### **Background:**

MP1B is a utility provided by IBM to analyze your IBM MQ environment's performance. MP1B shows you your SMF performance data and allows you to roll it off platform to CSV files for further analysis.

MP1B is installable at

[https://www.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~WebSphere&product=ibm/WebSphere/WebSphere+MQ&release=9.3.2.0&platform=z/OS&function=fixId&fixids=mp1b\\*](https://www.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~WebSphere&product=ibm/WebSphere/WebSphere+MQ&release=9.3.2.0&platform=z/OS&function=fixId&fixids=mp1b*)

Out of the box, it contains:

MQCMD – a program to display queue statistics and channel status over time

MQSMF – a program for interpreting your own accounting and statistics data

OEMPUT - a program to put/get messages in high quantities, useful for testing throughput

### **Overview of exercise:**

1. In this exercise, we will:
2. Use OEMPUT to populate a queue with a bunch of messages
3. Make sure settings are in place to record SMF data for our system
4. Run JCL to record our SMF data
5. Navigate the SMF data output to find performance problems in our queue
6. Interpret the performance problem

### **Exercise:**

1. MP1B has been installed on this environment, and you can find it by searching for the directory ZQS1.MP1B.JCL in the =3.4 data set search bar.

```

Menu  RefList  RefMode  Utilities  Help

Data Set List Utility

blank Display data set list          P Print data set list
V Display VT0C information          PV Print VT0C information

Enter one or both of the parameters below:
Dsname Level . . . ZQS1.MP1B.JCL
Volume serial . . .

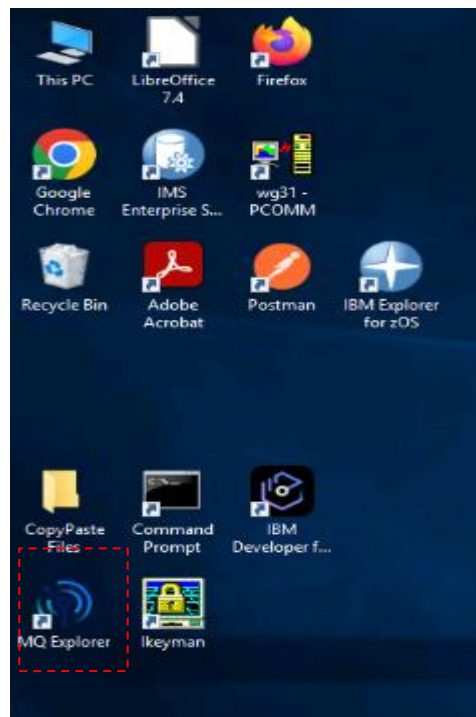
Data set list options
Initial View
1 1. Volume
2 2. Space
3 3. Attrib
4 4. Total

Enter "/" to select option
/ Confirm Data Set Delete
/ Confirm Member Delete
/ Include Additional Qualifiers
/ Display Catalog Name
- Display Total Tracks
- Prefix Dsname Level

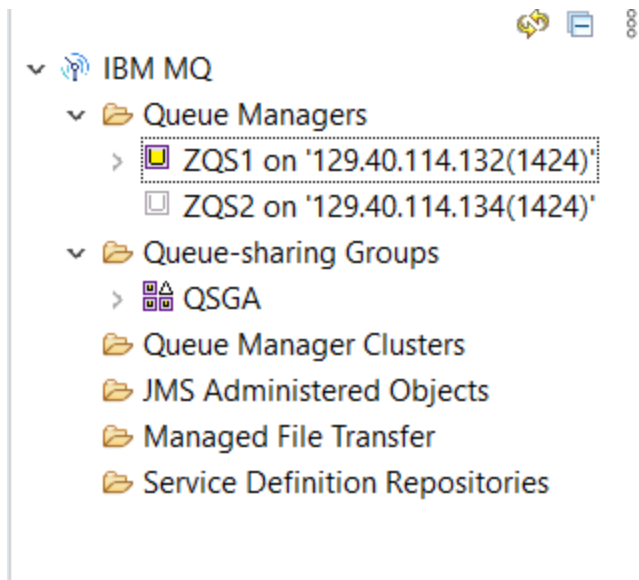
When the data set list is displayed, enter either:
Option ==>
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F12=Cancel

```

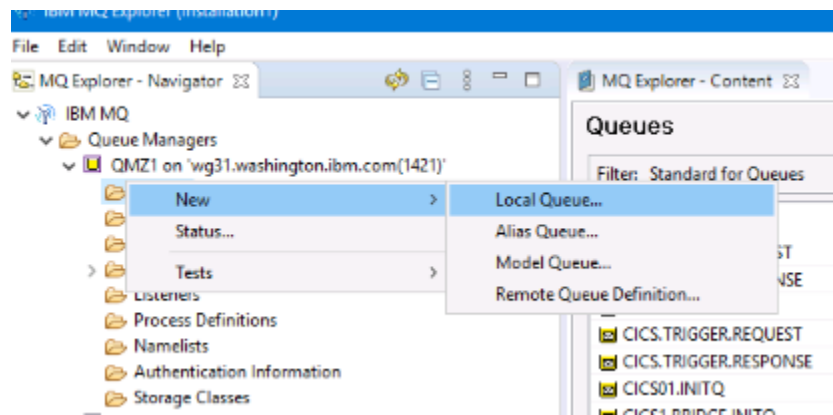
- Now, outside of z/OS, open up MQ Explorer on your Windows Desktop. The icon should look like this:



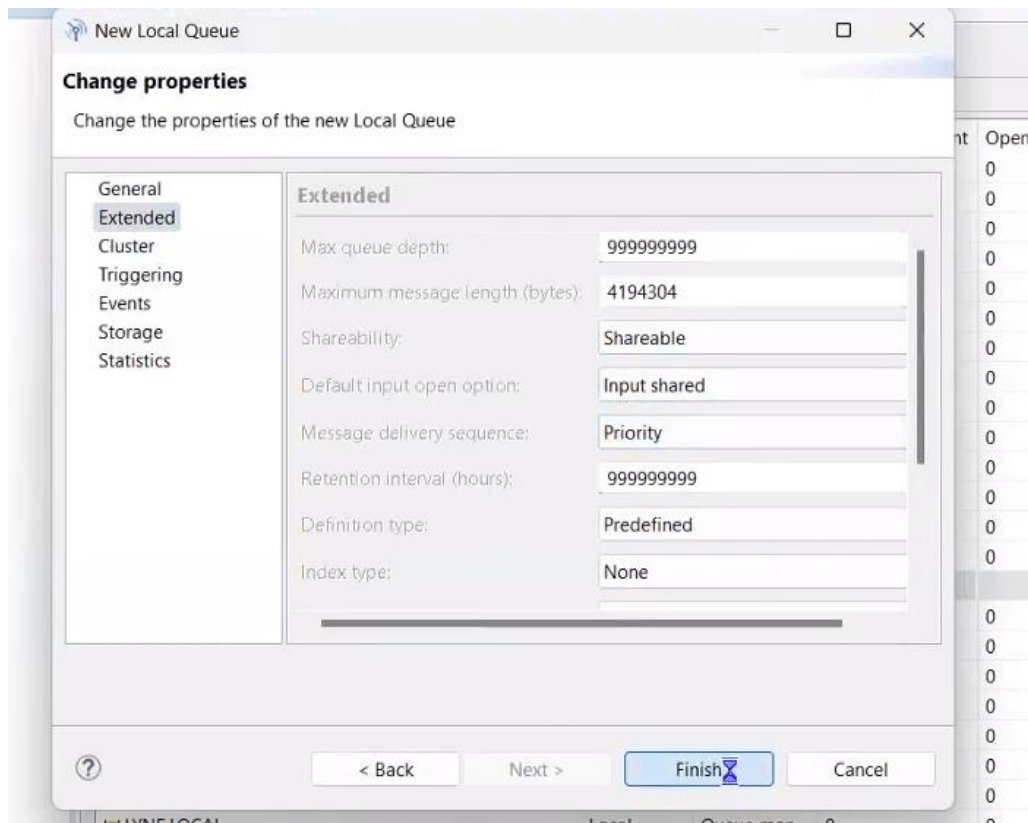
- Once you've opened MQ Explorer, you should see a left-hand menu bar like below. Right click on the ZQS1 queue manager and hit 'Connect'.



- By clicking on the arrow to the left of ZQS1, a dropdown list of MQ objects will appear. Right click on the 'Queues' folder and construct a new local queue called MP1B.TESTER.



5. Create a queue on your queue manager using MQ Explorer. The queue should have the following properties:



---

*Why make the queue shareable? Great question! Shareable queues tend to come in handy in a test environment, so that developers can browse the queues.*

---

6. Now that we have our queue defined, head back to z/OS. We will use OEMPUT to load messages into MP1B.TESTER. In the directory ZQS1.MP1B.JCL, place an 'e' to the left of the OEMPUT member.

```

000044 //*****
000045 //*
000046 //  SET QM=ZQS1
000047 //  SET Q=MP1B.TEST.QUEUE
000048 //S1  EXEC PGM=OEMPUT,REGION=0M,
000049 //  PARM=('-M&QM -tm3 -Q&Q -crlf -fileDD:MSGIN -P ')
000050 //SYSIN DD *
000051 /*
000052 //STEPLIB DD DISP=SHR,DSN=ZQS1.MP1B.LOAD
000053 //      DD DISP=SHR,DSN=MQ933CD.SCSQLOAD
000054 //      DD DISP=SHR,DSN=MQ933CD.SCSQAUTH
000055 //      DD DISP=SHR,DSN=MQ933CD.SCSQANLE
000056 //SYSPRINT DD SYSOUT=*
000057 //MSGIN DD DISP=SHR,DSN=ZQS1.MQ.JCL(MSGS)
000058 //

```

7. Once in OEMPUT, right submit on the command line and hit enter to load persistent messages into the queue manager.

---

a. *I won't summarize the whole JCL, but pay attention to this particular line: PARM=('-M&QM -tm3 -Q&Q -crlf -fileDD:MSGIN -P')*

b. *Lets break it down:*

c. *'-M&QM: queue manager name*

d. *-tm3: send messages for 3 minutes*

e. *-Q&Q the queue name*

f. *-crlf: each line in the input message file is used in sequence as message data*

g. *-fileDD:MSGIN: Use the MSGIN file as input*

h. *-P: Use persistent messages*

---

8. If you look at your MQ Explorer, you should now see that your queue is populated with lots of messages!

Current queue depth 289339
-------------------------------

9. Now, we will enter a series of MVS commands to adjust the settings of the queue manager to prepare it for the collection of SMF data. To do this, navigate to the ISPF main menu
10. Once in the ISPF main menu, enter 'd' in the command line and hit enter
11. Once in SDSF, place a / in the command input line and hit enter
12. A pop-up screen like this should pop up:

```

Edit  Options  Help
-----
                          System Command Extension
=====
=====
Comment                                STORELIMIT
Group                                Show *          (F4 for list)
=> display smf                                More:      +
=> zmq1 set system logload(200)
=> ZMQ1 DISPLAY SYSTEM
=> ZSHR STOP CHINIT
=> ZSHR STOP QMGR
=> display consoles
=> D CONSOLES
=> D OMVS.CINET

F5=FullScr F6=Details F7=Up F8=Down F10=Save F11=Clear F12=Cancel
SDSF
```

13. Enter the following commands here, one at a time. Each command will take you out of the System Command Extension window, so you will have to use the / command to return to the correct window for executing commands.
  - ZQS1 SET SYSTEM STATIME(1.00) to change the statistics time interval to 1 minute
  - ZQS1 SET SYSTEM ACCTIME(-1) to change the accounting time interval to match the statistics time interval
  - ZQS1 SET SYSTEM LOGLOAD(200) to change the log load attribute to the minimum.

---

*We want to modify our queue manager's log load attribute to be super low in order to manufacture a lot of checkpointing so we see something interesting in the SMF records for the purpose of the lab*

---

- DISPLAY SMF to see where SMF data is

---

*This tells us where our SMF data will be stored*

---

- ZQS1 ALTER QMGR STATCHL(MEDIUM)

---

*This tells z/OS we want to enable channel statistics to be collected at a moderate ratio of data collection*

---

- ZQS1 ALTER QMGR MONQ(MEDIUM)

---

*This tells z/OS to turn on monitoring for the queue manager's queues at a moderate ratio of data collection*

---

- ZQS1 ALTER QMGR MONCHL(MEDIUM)

---

*This tells z/OS to turn on monitoring for the queue manager's channels at a moderate ratio of data collection*

---

- ZQS1 START TRACE(STAT) CLASS(1,2,4,5)
- ZQS1 START TRACE(ACCTG) CLASS(3,4)

14. Now all the settings should be in place for our queue manager. Head back to ZQS1.MQ.JCL using 3.4 from the main ISPF menu.
15. Navigate to the SMFDUMP member. Once inside, enter 'submit' on the command line to execute SMFDUMP JCL. The SMFDUMP JCL starts with deleting old tasks, then outputs it in a specified location, in our case, MQADMIN.

```

000001 //ZQS2DSMF JOB
000002 /*JOBPARM SYSAFF=(MQS2)
000003 //DELETE1 EXEC PGM=IEFBR14
000004 //TASKDEL1 DD DISP=(MOD,DELETE),
000005 //          SPACE=(CYL,(100,20),RLSE),UNIT=SYSDA,
000006 //          DSN=ZQS2.QUEUE.MQSMF.SHRSTRM2
000007 //SMFDUMP EXEC PGM=IFASMF,REGION=0M
000008 //DUMPOUT DD DSN=ZQS2.QUEUE.MQSMF.SHRSTRM2,
000009 //          DISP=(NEW,CATLG,DELETE),
000010 //          RECFM=VB,BLKSIZE=27998,
000011 //          SPACE=(CYL,(100,20),RLSE),UNIT=SYSDA
000012 //SYSPRINT DD SYSOUT=*
000013 //SYSIN DD *
000014     LSNAME(IFASMF.DEFAULT,OPTIONS(DUMP))
000015     OUTDD(DUMPOUT,TYPE(115,116))
000016 /*
000017 //
Command ==>

```

16. You can check that the SMFDUMP is processing by navigating to your job using SDSF. Access SDSF using =D from the ISPF menu.
17. Once in SDSF, select ST from the menu and hit 'enter'
18. Type in 'prefix ZQS2\*'. This will show you a list of all jobs submitted that start with ZQS2. Remember, we define our job names at the top left of each JCL file.
19. Here, you put a '?' mark besides the jobname. Hit enter, then a screen with a SYSPRINT menu option should pop up. Next to SYSPRINT, put a 's' and hit enter.
20. Enter 'bottom' on the command line and you should see a screen like below, indicating that records are being written. You can also confirm this by looking in the output for the SUMMARY ACITIVITY REPORT.

SDSF OUTPUT DISPLAY ELKINSSG JOB17923 DSID	102 LINE 34	COLUMNS 02- 81
COMMAND INPUT ==> _		SCROLL ==> CSR
78 6 .18 %	3,028.00	2,000
88 48 1.45 %	243.68	161
89 6 .18 %	2,884.66	1,794
99 2,290 69.21 %	2,134.82	228
115 265 8.01 %	2,332.00	280
116 234 7.07 %	2,572.87	420
TOTAL 3,309 100 %	2,072.81	74
NUMBER OF RECORDS IN ERROR 0		
***** BOTTOM OF DATA *****		



21. After submitting, you will have to submit another job MQSMFP in ZQS1.MQ.JCL. This job will give us some formatted information about the SMF data

```

EDIT          ZQS1.MQ.JCL(MQSMFP) - 01.05                      Columns 00001 00071
***** Top of Data *****
==MSG> -CAUTION- Profile changed to CAPS OFF (from CAPS ON) because data
==MSG>          contains lower case characters.
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 //ZQS2MP1B JOB 'MP1B',NOTIFY=&SYSUID
000002 //*****
000003 /**
000004 /** <copyright
000005 /**      notice="copyright-lm-source"
000006 /**      pids="5655-MQ9"
000007 /**      years="2015,2023"
000008 /**      crc="2872671822" >
000009 /**
000010 /**      Licensed Materials - Property of IBM
000011 /**
000012 /**      5655-MQ9

```

22. Now, navigate to the SDSF output for the submitted job. We will be able to see the SMF output in useful categories that can also be exported as CSV files.

Display Filter View Print Options Search Help									
-----									
SDSF JOB DATA SET DISPLAY - JOB ELKINSSF (JOB17924)						DATA SET DISPLAYED			
COMMAND INPUT ==>						SCROLL ==> CSR			
NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cnt	Page
	JESMSGLG	JES2		2	ELKINSC	0	LOCAL	26	
	JESJCL	JES2		3	ELKINSC	0	LOCAL	80	
	JESYSMSG	JES2		4	ELKINSC	0	LOCAL	185	
	SYSPRINT	S1		103	ELKINSC	0	LOCAL	112	
	ADAP	S1		106	ELKINSC	0	LOCAL	420	
	ADAPCSV	S1		107	ELKINSC	0	LOCAL	31	
	BUFF	S1		108	ELKINSC	0	LOCAL	663	
	BUFFIO	S1		109	ELKINSC	0	LOCAL	48	
	BUFFCSV	S1		110	ELKINSC	0	LOCAL	122	
	CF	S1		111	ELKINSC	0	LOCAL	26	
	CFCSV	S1		112	ELKINSC	0	LOCAL	11	
	CHINIT	S1		113	ELKINSC	0	LOCAL	305	
	CHINCSV	S1		114	ELKINSC	0	LOCAL	31	
	CMESSAGE	S1		115	ELKINSC	0	LOCAL	110	
	DATA	S1		116	ELKINSC	0	LOCAL	211	
	DB2	S1		117	ELKINSC	0	LOCAL	42	
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK									
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
*SDSF DSLIST -DSLIST									

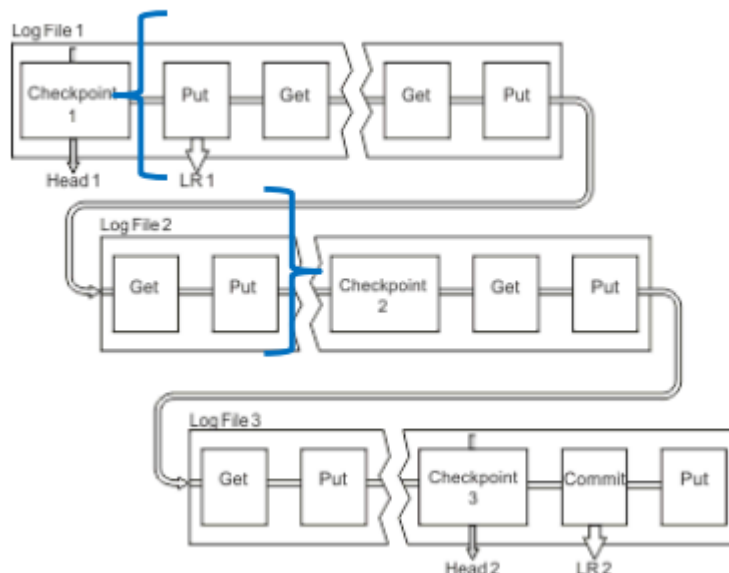
23. Navigate to the LOG statistics by putting a 's' next to it and hitting enter. Scroll down until you see a screen similar to the one below.
24. Here you can see LLCheckpoints has a value of 1564. Within our interval, we would expect this value to be 0's or single-digits. 1564 is way too high. This indicates we should adjust our LOGLOAD attribute to have it write more log records between checkpoints.

```

Write_Wait      0, Write_Nowait    311222, Write_Force      1564,
Read_Stor       4, Read_Active     0, Read_Archive     0,
BSDS_Reqs      3157, CIs_Created   28743, BFWR             172880,
ALW             0, CIs_Offload     0, LLCheckpoints    1564
Read_delayed    0, Tape_Lookahead  0, Lookahead_Mount  0

```

### Summary:



The LOGLOAD parameter specifies the number of log records that are written between checkpoints. In the figure above, you can see the LOGLOAD indicated by the blue brackets. For the above image's example, the LOGLOAD looks to be 6 here (6 would be impossibly small in a real environment).

We set our queue manager's LOGLOAD attribute to the lowest possible value of 200 then flood our environment with messages. We saw see this cause high checkpointing in our recorded SMF window, resulting in unnecessary consumption of processor time and additional I/O.

