MQPERF1

Introduction to MQ SMF- Overview, Data Capture and Initial Post Processing

Your first MQ SMF experience

Agenda

- Background
- Setting up for capture
- MQ SMF – Processing the data
  - We have to use what is free
- The statistics data
- The accounting data
- Additional information

Background & Objectives

- MQ does not report into the z/OS Health checker
  - So how do I know if my queue manager is healthy?
  - You have to use a combination of sources
    - The MQ JES log for storage use, the number of log switches, etc.
    - The MQ SMF data and general RMF data to see how the resources are really being used
  - From active MQ for z admins, 'I spend 50% of my time or more proving that MQ on z is NOT the problem.'

- SMF data doesn't lie
  - Though it can be misinterpreted and under captured!

Mea Culpa

- Because of the continued use of MQSMFCSV I have discovered some errors in the collection, production, and in my understanding and interpretation of some of the MQ SMF data.

- You will see some new rules and explanations where I have found these issues.

- I once again owe a great deal of thanks to Gwydion Tudur and Mark Taylor for investigating these issues and responding after their investigation of the actual code.

Types of MQ SMF and Collection Classes

- The MQ SMF data is broken down into two major categories (and SMF data types)
  - Statistics – Type 115
  - Accounting – Type 116
- The generation is controlled by type and class
  - Do not confuse the classes with the subtype
  - Statistics classes
    - * - collect all classes (01,02) of statistics except channel initiator
    - 04 – channel initiator statistics class
  - Accounting Classes
    - * - collect all classes of accounting data except channel accounting
    - 01 – only collect the QMAC data, only used for chargebacks
    - 03 – Collect the Task with queue use accounting
    - 04 – channel accounting records

MQ Statistics – The basic health of the QMGR

- The SMF 115 data is the statistical information produced by a IBM MQ for z/OS queue manager.
  - Primarily used to track major trends and resolve performance problems with the queue manager
  - Very lightweight
    - At least two records per queue manager per SMF interval (V8+)

Statistics Data – Source and Subtype

Source	Subtype
Storage Manager	1
Log Manager	1
Message Manager	2
Data Manager	2
Lock Manager	2
Db2 Manager	2
Coupling Facility Manager	2
Topic Manager	2
SMDS Usage	2
Buffer Manager	215
Channel Initiator	231
Data Manager – Page Set	201

MQ Accounting – The Task and Queue details

- The SMF 116 data is the accounting information produced by a IBM MQ for z/OS queue manager.
  - Primarily used to determine what is going on within IBM MQ workload
  - Heavyweight
    - Very much so!
  - Individual tasks get multiple large records produced
    - Each task gets records produced at the end of the task
    - Long running tasks (like channels, batch jobs, long CICS reader transactions) will get multiple sets of task records at each SMF interval
  - Channel accounting records are accumulated and produced at SMF intervals (not when the channel stops)

MQ Accounting – The home of the devil

- The SMF 116 data is the task related data produced by an IBM MQ for z/OS queue manager.
  - Very detailed
  - Often necessary to track down performance problems
  - Costs vary by:
    - Application Style
    - SMF production type (MAN datasets or Logstreams)
    - Recently seen some examples of there being little to no overhead for collection and production of the data
    - Standard estimates are between 3-7% overhead

Accounting Data – Source and Subtype

Source/type of data	Subtype	Comments
Message Manager	0	The 'QMAC' records at times used for chargebacks, largely ignored these days
Thread identification record	1	Task ID
Thread accounting	1	Task accounting info – things not associated with an individual queue
Queue Accounting	1	Queue use for this task
Thread identification record	2	Task ID - overflow
Queue Accounting	2	Queue use overflow
Channel Accounting	10	Individual channel accounting records

Capture

Setting up for Capture

- CSQ4ZPRM
  - SMFSTAT=NO – Default, (ARRGGGHHH!) should be changed to SMFSTAT=(*)
    - Gathering and producing the statistics is not expensive
    - Most are always gathered, just written when the interval expires
    - **NEW Knowledge**
      - **Using the asterisk does not include the class 4 data.**
      - **AND if you have SMFSTAT=NO, you must turn on all the classes first and then turn on the class 4 data collection independently.**
  - SMFACCT=NO – Default, normally controlled via commands
  - STATIME – the interval, in minutes, between the creation of the SMF statistical and long running task accounting records
    - 30 – default, every 30 minutes
    - 0 – Use the system wide SMF interval, usually preferred
    - Any other integer up to 1440
      - Once a day

Setting up for Capture - continued

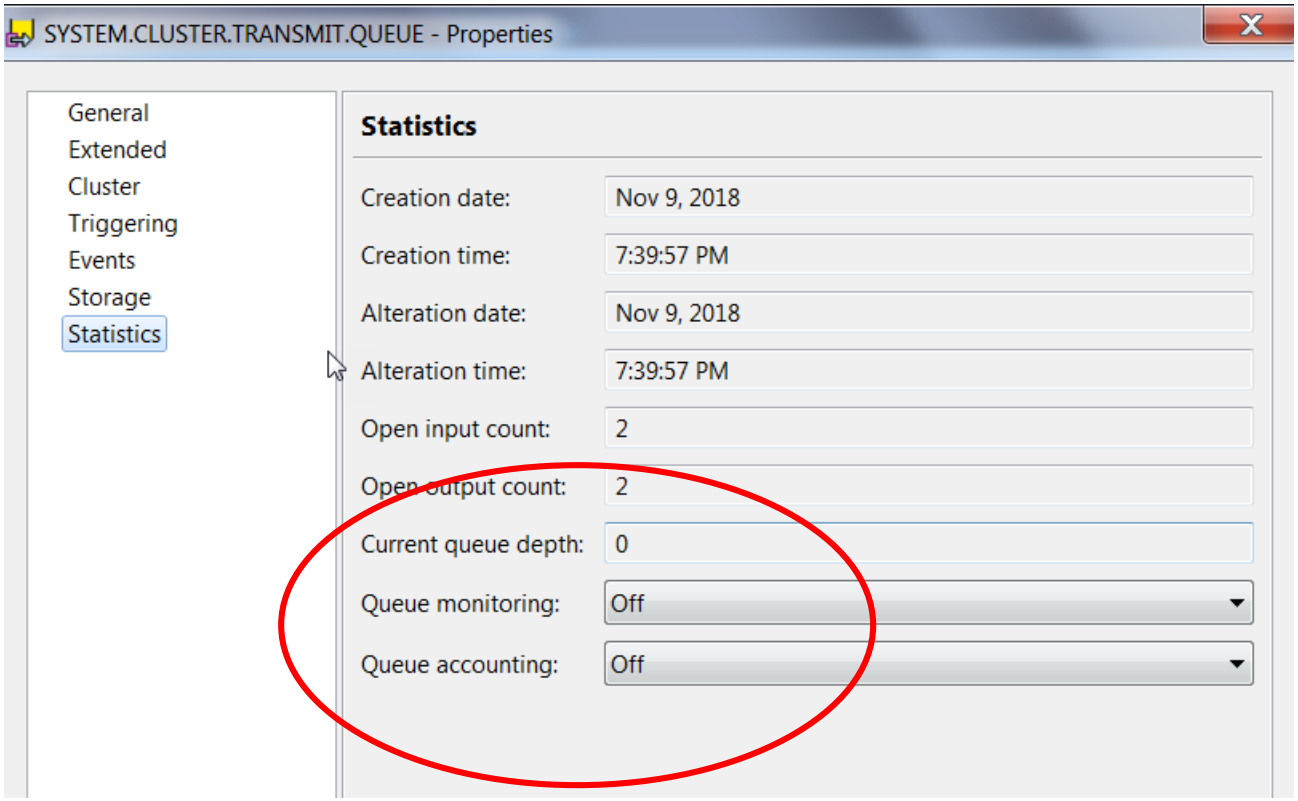
- **START TRACE Command**
  - START TRACE(S) CLASS(*)
    - Starts the statistics production for the queue manager
    - Note that if you have never produced this data, the first record should be ignored. It will have data from when the queue manager started.
  - START TRACE(S) CLASS(4)
    - Starts the channel initiator statistics
  - START TRACE(A) CLASS(*)
    - Starts the task accounting capture and production
    - Note that tasks that cross interval boundaries will cut a set of accounting records per interval reflecting the activity for that interval
  - START TRACE(A) CLASS(4)
    - Starts the channel accounting trace

Setting up for Capture - concluded

- SET SYSTEM STATIME (interval)
  - The interval is in minutes
  - Change takes effect at the end of the current interval
    - So if you've been silly and set it to a full day (1440), it will be a day before this takes effect
  - Often used to shorten the interval when trying to isolate a performance problem.

Check your queue definitions!

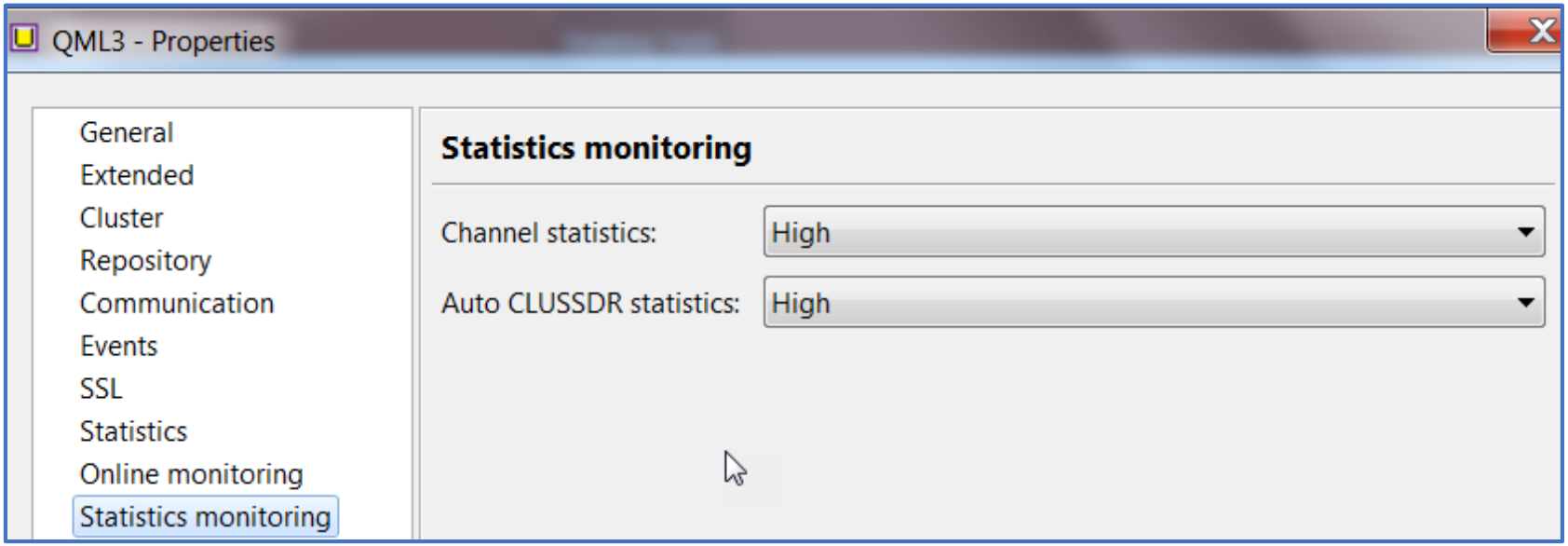
- This makes me a bit crazy:



Check you QMGR definitions!

```
 BROWSE     MQ910.SCSQPROC(CSQ4INYG)
 Command ===>
*           MONQ( OFF )  +
*           MONCHL( OFF )  +
*           MONACLS( QMGR )  +
*           STATCHL( OFF )  +
*           STATACLS( QMGR )  +
```

These QMGR values should look like this



Post-processing the data – MQ Extraction

- SMF data can be on z/OS Logstreams or written to the 'MAN' datasets
- The MQ Data can be extracted using the IFASMFDP program (datasets) or IFASMFDL (log streams)
- The options are similar for both extraction program, but the important one is the TYPE option:
  TYPE(115,116)
  - That will pull the MQ statistics and accounting data of all subtypes
- At some point we plan to test real streaming the SMF data into MQSMFCSV, but we are not there yet!

Sample SMF Extract

```
//SMFDUMP8 EXEC PGM=IFASMFDP
//DUMPINA  DD   DISP=SHR,DSN=MPXCAT.SMF.MPX1.MANY
//DUMPOUT  DD  DSN=ELKINSC.TESTPUT1.MQSMF01,
//             DISP=(NEW,CATLG,DELETE),
//             VOL=SER=Q70006,
//             RECFM=VB,BLKSIZE=27998,
//             SPACE=(CYL,(10,10),RLSE),UNIT=SYSDA
//SYSPRINT DD   SYSOUT=*
//SYSIN  DD *
  INDD(DUMPINA,OPTIONS(DUMP))
  OUTDD(DUMPOUT,TYPE(115,116))
/*
```

Additional Information

- Info on the Dataset dump program:
  https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ieag200/dump.htm

- The dataset dump formatter options:
  https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ieag200/dumpopt.htm

- Info on the Logstream dump program:
  https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ieag200/dmplog.htm

- The Logstream dump formatter parameters:
  https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ieag200/dumplopt.htm

So I’ve gotten the data, what do I do with it?

- Well that depends…

MQ SMF – Processing the data

- CSQ4SMFD
  - Provided with MQ
  - Dump format of the data

```
                                                                    CSR
******************************** TOP OF DATA ********************************
 message manager statistics data
--Q-M-S-T---H-E-X---P-R-I-N-T----
Address  = 2072AC08
00000000 : D40F0048 D8D4E2E3 000024FE 00002402 ⟨M...QMST.....⟩
00000010 : 0000EB1A 0000B480 00000000 00000C48 ⟨................⟩
00000020 : 00000000 00000000 00000000 00000000 ⟨................⟩
00000030 : 00000000 00000000 000000B4 00000438 ⟨................⟩
00000040 : 00000000 00000000                   ⟨........        ⟩
--Q-M-S-T---F-O-R-M-A-T-T-E-D----
qmstid   = d40f
qmstll   = 0072
qmsteyec = QMST
qmstopen = 00009470
qmstclos = 00009218
qmstget  = 00060186
qmstput  = 00046208
```

MP1B – the standard post processor

- Provided as an IBM MQ SupportPac
- The MQSMF post processor may be used to format the MQ SMF data, and can assist with both monitoring the health of the z/OS queue managers and with problem determination.
- The reports and CSV files can help with these tasks:
  - Determine what applications are using a particular queue manager and it's objects
  - Listing the queues that are actually in use
  - Locating the storage used by particular queues
  - Which system resources may be under strain
  - Which MQ verbs are being used by individual tasks
  - I/O responsiveness issues
  - etc

MQ SMF – Processing the data

- MP1B – MQ SMF report formatter
  - Message Manager, MSGM output file, report sample:

```
Message Manager
MPX2,QML2,2017/01/12,08:50:54,VRM:800,
  From 2017/01/12,08:21:04.455699 to 2017/01/12,08:50:54.787219, duration  1790 seconds.
   MQOPENs      9470,  MQCLOSEs      9218,  MQGETs     60186,  QMLUTs     46208
   QMLUT1s         0,  MQINQs       3144,  MQSETs         0,  C ALL H        0
   MQSUBs          0,  MQSUBRQs        0,  MQCBs        180
   MQCTLs       1080,  MQSTATs         0,  Publish        0
   MQGet rate 33.000000/sec  QMLut rate 25.000000/sec
MPX1,QML1,2017/01/12,08:53:53,VRM:800,
  From 2017/01/12,08:24:00.111232 to 2017/01/12,08:53:53.553654, duration  1793 seconds.
   MQOPENs     27170,  MQCLOSEs     24670,  MQGETs    325273,  QMLUTs    291386
   QMLUT1s         0,  MQINQs       1043,  MQSETs         0,  C ALL H        0
   MQSUBs          0,  MQSUBRQs        0,  MQCBs       2958
   MQCTLs       8933,  MQSTATs         0,  Publish        0
   MQGet rate 181.000000/sec  QMLut rate 162.000000/sec
```

MQ SMF – Processing the data

- MP1B – MQ SMF report formatter
  - Message Manager CSV, MSGMCSV output file, sample:

```
MVS,QM,Date,Time,Puts,Put1s,Gets,Open,Close,Inquire,Set,"Close all H",Sub,SubR,"Reg CB",Control,Stat,Publish,
MPX1,QML2,2017/01/12,08:50:54,     46208,         0,     60186,      9470,      9218,      3144,         0,         0,         0,
      0,       180,      1080,         0,         0,
MPX1,QML1,2017/01/12,08:53:53,    291386,         0,    325273,     27170,     24670,      1043,         0,         0,         0,
      0,      2958,      8933,         0,         0,
MPX1,QML3,2017/01/12,08:54:25,     40824,         0,     44904,      1735,      1700,       232,         0,         0,         0,
      0,        57,       440,         0,         0,
MPX1,QML4,2017/01/12,08:55:10,     44112,         0,     48907,      2008,      1910,       337,         0,         0,         0,
      0,        57,       427,         0,         0,
```

MQ SMF – Processing the data

- **MP1B is a great place to start with examining the data**
  - If you have not looked at the data previously, it gives guidance on where you should look for potential problems
  - In the WSC we use this to give us a “first look”

MQ SMF – Processing the data

- MP1B produces messages to warn of potential problems

```
MQQPST02S MPX1,QML1,2017/01/10,23:31:36,VRM:800, BP 1 Filled many(386) times.    This is typical of long lived messages. Buffer pool
 may be too small
MQQPST04E MPX1,QML1,2017/01/10,23:31:36,VRM:800, BP 1 Many (626104) pages read from disk.  This is typical of long lived messages. B
uffer pool may be too small
MQQJST00I MPX1,QML1,2017/01/11,00:01:37,VRM:800, Log read log buffers from storage 14 > 0
MQQJST01W MPX1,QML1,2017/01/11,00:01:37,VRM:800, Log read log buffers from active logs 458 > 0
MQQJST03E MPX1,QML1,2017/01/11,00:01:37,VRM:800, Log Number of checkpoints 24 > 10
MQQJST11W MPX1,QML1,2017/01/11,00:01:37,VRM:800, logging rate is low  14 < 50 MB/Sec
MQQIST01W MPX1,QML1,2017/01/11,00:01:37,VRM:800, QIST read ahead message count  285 > 0
MQQIST02W MPX1,QML1,2017/01/11,00:01:37,VRM:800, QIST Message read from disk 122 > 0
MQQPST00W MPX1,QML1,2017/01/11,00:01:37,VRM:800, BP 1 Many(1802) buffers written immediately.  Buffer pool may be too small
MQQPST02S MPX1,QML1,2017/01/11,00:01:37,VRM:800, BP 1 Filled many(339) times.    This is typical of long lived messages. Buffer pool
 may be too small
MQQPST04E MPX1,QML1,2017/01/11,00:01:37,VRM:800, BP 1 Many (391542) pages read from disk.  This is typical of long lived messages. B
uffer pool may be too small
MQQJST01W MPX1,QML1,2017/01/10,20:32:38,VRM:800, Log read log buffers from active logs 93 > 0
```

Additional formatting tools - Github project

http://github.com/ibm-messaging/mq-smf-csv

We will talk more about this tomorrow!

Commercial Interpreters

- There are several available on the market.

Finally – Making things more manageable

- The SMF dump utilities are not very granular
  - And when faced with millions of records, that can be daunting
- Use DFSORT to extract data for
  - Specific queue managers
  - Specific batch jobs
  - Specific transactions

Sorting for CICS TRANID

```
//*
//*   SORT BY THE CICS TRANID
//*
//SYSIN   DD *
  OPTION VLSHRT
  SORT FIELDS=(109,4,BI,A)
  INCLUDE COND=(109,4,CH,EQ,C'ABCD')
/*
```

Summary

- The SMF data can be used in many ways to find patterns of use, problems with the queue managers, and application programming problems.

- There are many other things within the data that are helpful, and more to come with V9 and the formatting programs.