# Gathering the MQ SMF

**Lyn Elkins –**
**elkinsc@us.ibm.com**

# Agenda

- MQ SMF and Collection Classes
- Setting up for capture
  - The zPRMs
  - The Commands
- Closing

# 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
    - Classes 01 and 02 are the queue manager statistics
    - Class 04 is the channel initiator statistics
  - Accounting Classes
    - Class 01 – the QMAC data, pre V6 data, only used for chargebacks
    - Class 03 – the Task accounting data, that includes the queue use (by the data) information
    - Class 04 – the channel accounting data

# MQ Statistics – The basic health of the QMGR

- The SMF 115 data is the statistical information produced by an 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 – Lots of data!

- 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 – Cost of Collecting

- How much is this collection going to cost me?
  - 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
  - Costs will be higher if this is an emergency

# 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 – z Parms

- CSQ4ZPRM
  - 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
  - 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
  - SMFACCT=NO – Default, normally controlled via commands

# Setting up for Capture - Commands

- +cpf 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.
- Recommendations:
  - For normal capture set the value to 0
    - Allows coordination with other SMF/RMF capture
  - For performance issues and problem determination set to 5 or less

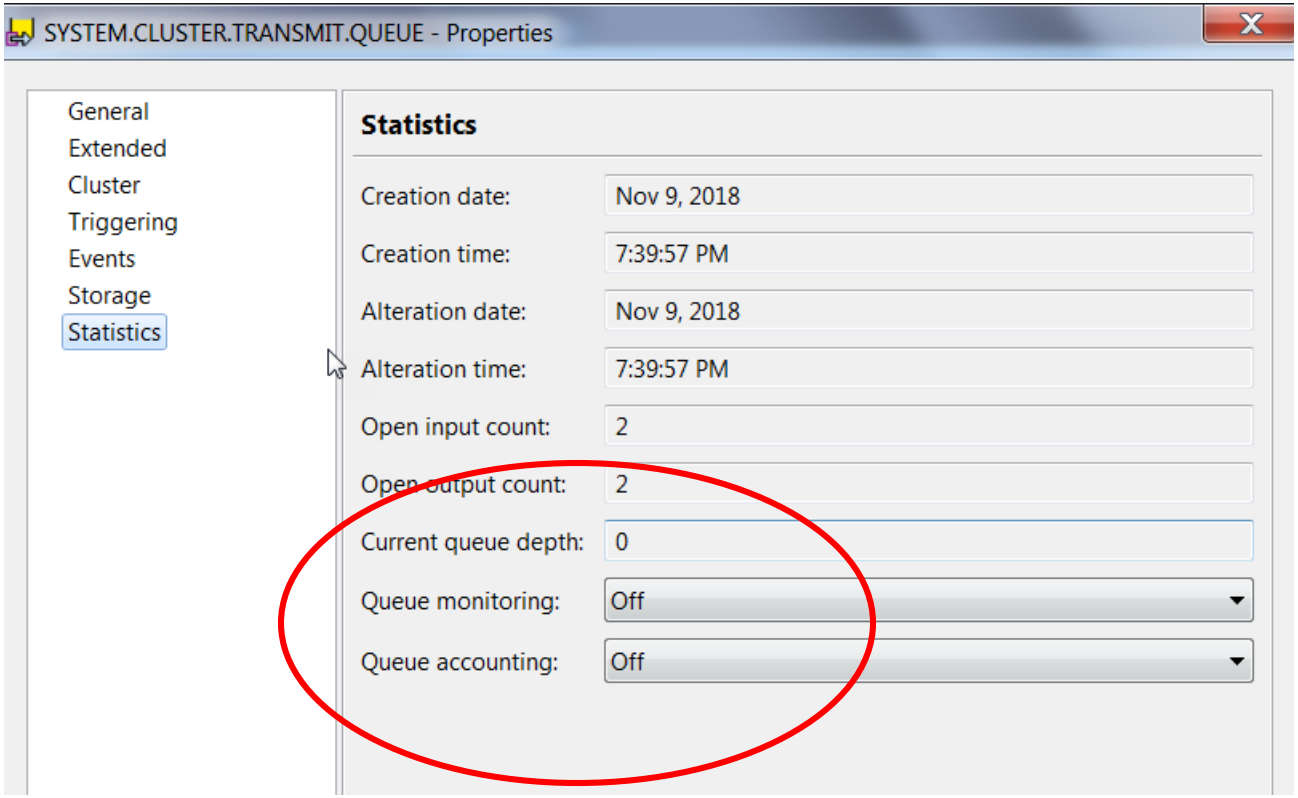
# Setting up for Capture - continued

- **START TRACE Command**

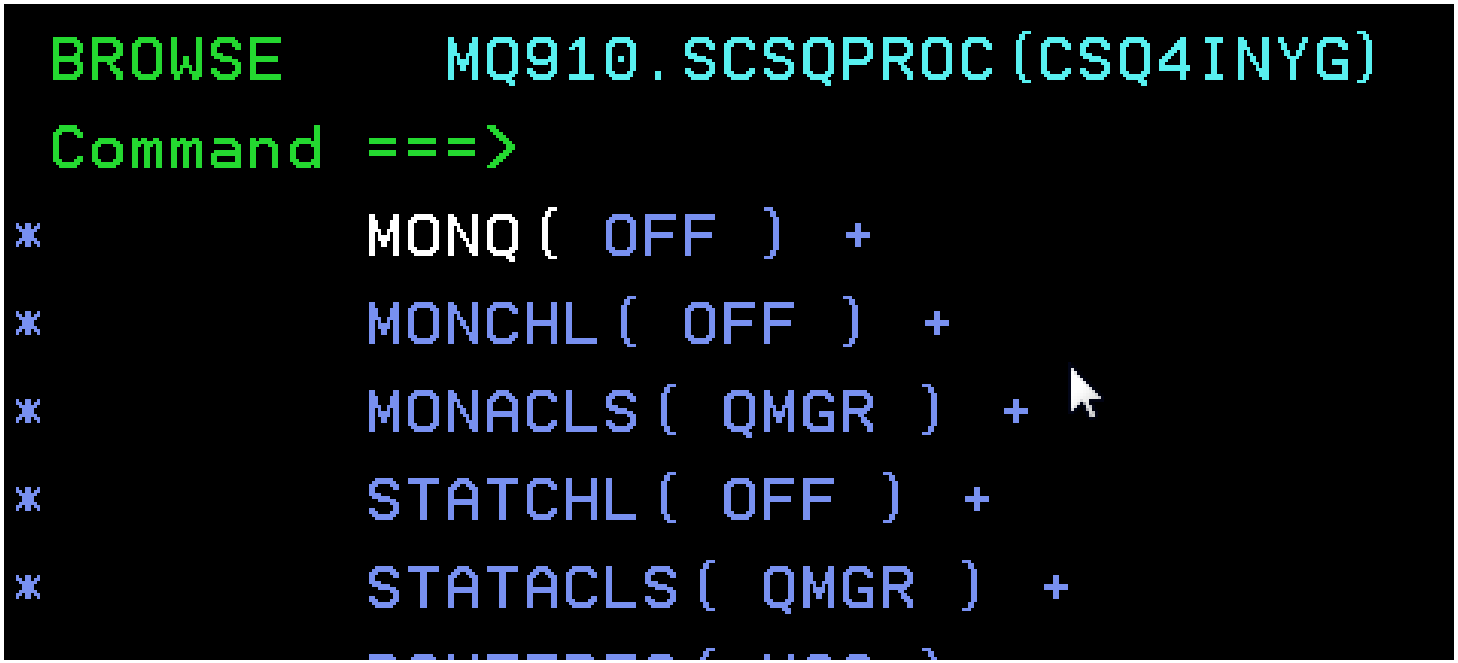
- +cpf START TRACE(S) CLASS(*)
  - Starts the statistics production for the queue manager
  - Note that if you have never produced this data, the first record should be ignored. It will have data from when the queue manager started.
- +cpf START TRACE(S) CLASS(4)
  - Starts the channel initiator statistics
- + cpf 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
- +cpf START TRACE(A) CLASS(4)
  - Starts the channel accounting trace

# Check your queue definitions!

- This makes me a bit crazy:



# Check you QMGR definitions!



# Summary

- Capturing the SMF data is the beginning of the story.
  - I'll move on to processing the data soon.