

IBM MQ - What's New in Logging in V9.1?

Christopher Frank

chrisfra@us.ibm.com

MQ 9.0.x CD content, available with V9.1 LTS

Replicated Data
Queue Manager
for MQ
Advanced

Linear logging
automation and
performance

RESTful
administration

Error log
formatting

Web Console

RESTful
messaging

MQ Appliance
SAN support

MQ JMS in
CICS Liberty
Profile

Salesforce
bridge

AMS
confidentiality
performance on
z/OS Advanced

Blockchain
bridge for MQ
Advanced

Floating IP
support for MQ
Appliance

Code repository
integration

Backup and
Restore on MQ
Appliance

Redistributable
MFT agent for
MQ Advanced

Enhanced MFT
diagnostics

Cross LPAR
MFT agents for
z/OS Advanced

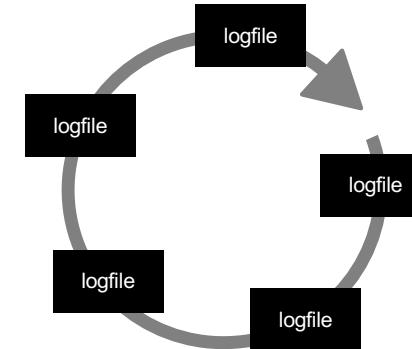
SNMP and
REST support
for MQ
Appliance

Types of Logging

MQ always logs all the data you need to recover from a queue manager failure in a recovery log

Restart recovery (circular and linear logging)

Enough information held in the log files to rebuild MQ resources to the level that they were at prior to the queue manager stopping



Circular logging

Media recovery (linear logging only)

Enough information held in the log files to rebuild MQ resources in the event of losing or corrupting MQ data

Linear logging comes with an added cost in terms of performance and administrative effort



Linear logging

What's Changed?

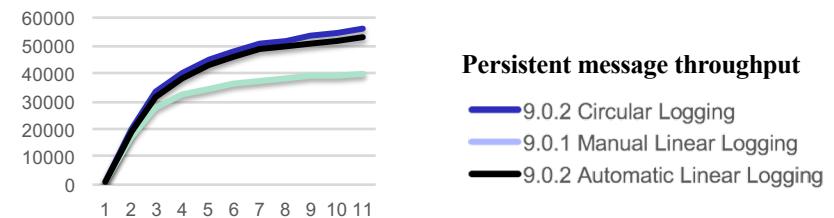
Automatic media imaging (Linear logging)

Media images can now be automatically scheduled by the queue manager, simplifying the administrative tasks and smoothing out the performance impact, simplifying the problem of when to take an image



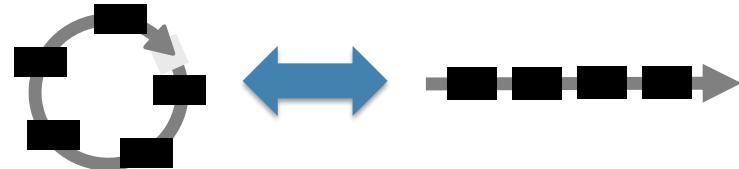
Automatic log reuse (Linear logging)

Constantly creating new linear logs reduces MQ's performance. Logs can now be reused by a queue manager to regain that performance. Choosing automatic reuse removes another administrative task of constantly deleting linear logs



Migrating between linear and circular

MQ now makes it possible to migrate a queue manager from linear to circular logging and vice versa



Additional Goodies

A few miscellaneous changes and new features that might take a bit of digging to find



Automatic Media Imaging

Media Recovery

- One key difference between Circular vs Linear logging is *Media recovery*
 - Can restore a damaged MQ object from an image in the log
 - Can be used to recover any MQ object
 - But particularly useful when that object contains business data
- Media recovery requires that images be captured periodically
 - This has always been an administrative responsibility
 - Images should be captured:
 - Frequently enough to make the recovery time satisfactory
 - Not so frequently that it degrades messaging performance

Why Automate Media Images?

- Writing media images to the log can dramatically impact MQ performance
- Some times are good for taking a media image:
 - When a queue is empty
 - When the system is quiet
 - When the size of the log required for media recovery is large
 - When a lot of time and/or activity has passed since the last one
- Other times are not so good:
 - When the queue is large or growing
 - When a lot of activity is taking place in the system
 - When an image was just taken, even if the queue is empty
- Careful scheduling of images needed
 - Easier said than done – it can be very difficult to take images at only the good times

How the Queue Manager can help

- A queue manager is constantly aware of its state. It knows things like:
 - How many messages are on each queue
 - When and how an object is changed
 - When the queue manager is busy (and not busy)
 - When the last image was taken
- The queue manager can choose the best time to record an image
- But it need guidance:
 - How long is an acceptable time between images?
or
 - What size to aim for the amount of media recovery data?
- With some target values, coupled with its knowledge of the state of the system, a queue manager could decide for itself the best time to take a media image

New Media Imaging Target Values

- MQ V9.1 adds a new QMGR attribute: **IMGSCHED**
 - *MANUAL* means image recording remains an administrative task
 - *AUTO* means the queue manager will handle the recording of images
 - You can still take manual images if you want to
- If *AUTO* specified, two additional QMGR attributes specify the imaging target values:
 - **IMGINTVL**: This is the target frequency for recording media images
 - Default = 60 minutes
 - *OFF* indicates that automatic media images will not be recorded on a time interval basis
 - **IMGLOGLN**: This is the target amount of log written after which media images will be taken
 - *OFF* (the default) indicates that automatic media images will not be recorded based on the number of bytes written

Media Imaging is Smarter as a Result!

- MQ uses the targets you supply to be clever about taking media images
- When you take a media image is taken manually, it is of the object in its current state
- When MQ does so automatically, it uses several strategies
 - For queues that are busy, MQ might take a *partial image* of it
 - An image of the queue as it was at a point slightly in the past.
 - Messages put or got to the queue since the recovery point do not have to be written to the image
 - Since recovering the queue would replay all log records and so would get reconciled then
 - Recording the image would be significantly faster using this strategy
 - The trade-off being that replaying would take a little more time.
 - This strategy works best when most messages sit on the queue briefly
 - Applications that use MQ as a database will subvert this
- Best Practice for media images is to take them when queues are empty or nearly so
 - So the queue manager watches for such times and records the image at an opportune time
 - It is much easier for the queue manager to spot such times than for an MQ admin

Being Selective about what is Recoverable

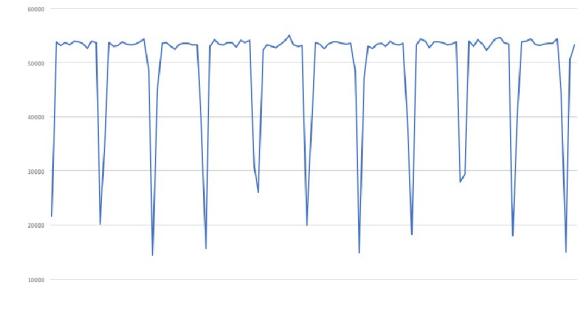
- Media images allow you to recover damaged objects
- But many objects may never need to be recovered
 - It may be less costly to delete and recreate some objects from a saved definition
 - e.g. Topic objects: Unlike queues they have no associated data – only attributes
- You can now choose whether local queues and other objects are recoverable
 - If you choose for them to be not recoverable then you cannot record images or recover them
 - And, automatic media images will not be taken for such objects
 - But less data will need to be written to the log
 - And those objects can be saved and restored – just by other means (e.g. dmpmqcfg)

Controlling Recoverability

- MQ V9.1 adds new QMGR attributes to control whether objects are recoverable or not:
- **IMGRCOVO**: Determines whether various object types are recoverable
 - *AUTHINFO, CHANNEL, CLNTCONN, LISTENER, NAMELIST, PROCESS, QALIAS, QREMOTE and SERVICE* objects
- **IMGRCOVQ**: Specifies the default *IMGRCOVQ* attribute for local and PERMDYN queue objects
 - Both can be set to YES or NO (Default is YES)
- A queue attribute is also defined for local and model queues:
 - **IMGRCOVQ**: Determines whether a local or PERMDYN queue object is recoverable
 - Can be set to YES, NO or QMGR (Default)

Value of Automatic Media Imaging

- Queue manager controlled imaging reduces the impact on other workload during recording
- Charts illustrate this:
 - At top: Manually imaging a deep queue
 - Bottom: Auto imaging of the same queue
- Impact on messaging workload is significantly reduced
 - Same data being recorded in the same span of time
 - But images records interleaved
 - Writing is cleverly scheduled
 - I/O avoidance, etc
- Application impact minimized
 - With no loss of recoverability



Manually controlled recording



Queue manager controlled recording

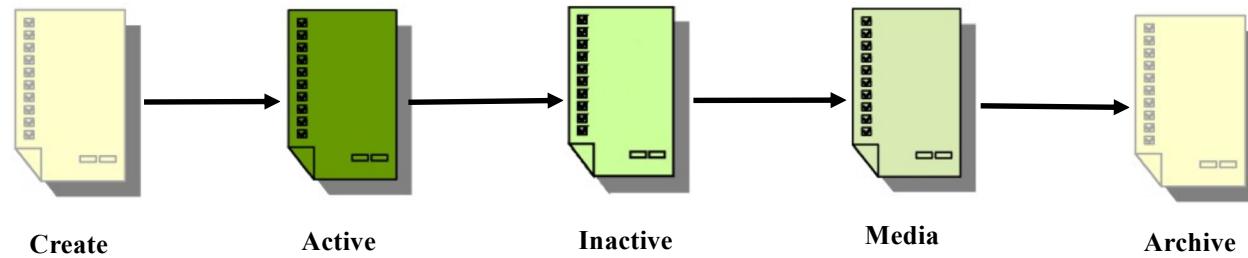
Questions?



Automatic Reuse of Log Files

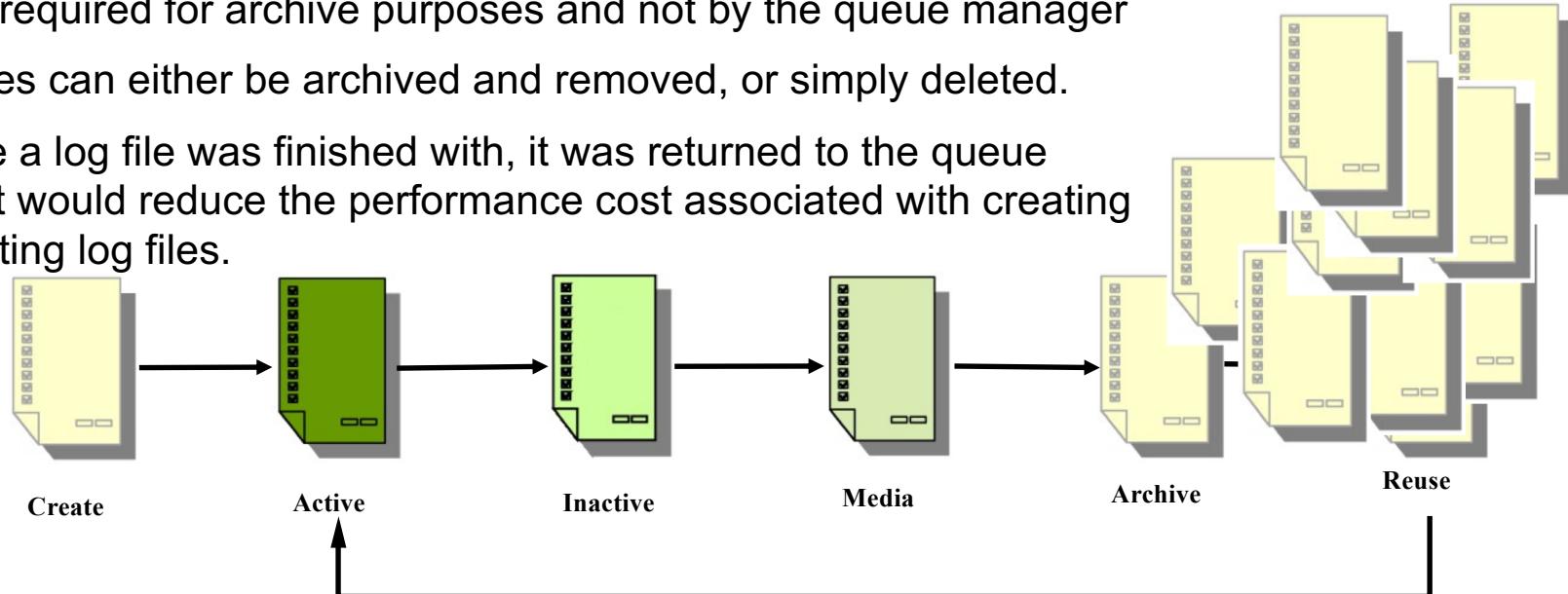
The Life Cycle of a Linear Log File

- Created and formatted by the queue manager
- Becomes part of the primary log (active)
- Once fully written it becomes (logically) read only and inactive
 - It may still be required for restart recovery
- Eventually it ceases to be required for restart recovery and required only for media recovery
- After this it is no longer needed by the queue manager and the customer can do whatever it likes with it



Reusing Old Log Files

- It is the responsibility of the customer to ensure that old logs are dealt with in a timely fashion
 - If they are not, the file system will eventually become exhausted
- Once a log file is no longer needed for restart or media recovery it is only potentially required for archive purposes and not by the queue manager
 - These files can either be archived and removed, or simply deleted.
- But if, once a log file was finished with, it was returned to the queue manager, it would reduce the performance cost associated with creating and formatting log files.



New Log Management Modes for Linear Logging

- MQ V9.1 introduces three modes for Linear Log Management:
 - **Manual** – Log extents must be manually managed, as they are today
 - **Automatic** – The queue manager will automatically reuse log extents when they are no longer required for restart or media recovery
 - **Archive** – Same as Automatic, except that the queue manager will wait for notification that an extent has been archived before reusing an extent

Setting Log Management

- When you create your queue manager you can specify new options:

- **-lla** Indicates that Automatic log management will be used

```
# crtqm -lla AUTO_LOG  
IBM MQ queue manager created.  
...
```

- **-lln** Indicates that Archive log management will be used

```
# crtqm -lln ARCHIVE_LOG  
IBM MQ queue manager created.  
...
```

- **-ll** Indicates that Manual log management will be used, as before

- If you have an existing queue manager you can edit qm.ini and can simply add one of the following lines to the Log stanza:

- LogManagement=Automatic or
 - LogManagement=Archive

```
qm.ini (for example)  
Log:  
LogManagement=Automatic
```

Archive Log Management

- When using Archive log management you will need to notify the queue manager when a log extent has been archived
- This is done using the new SET LOG command, or its PCF equivalent. For example:

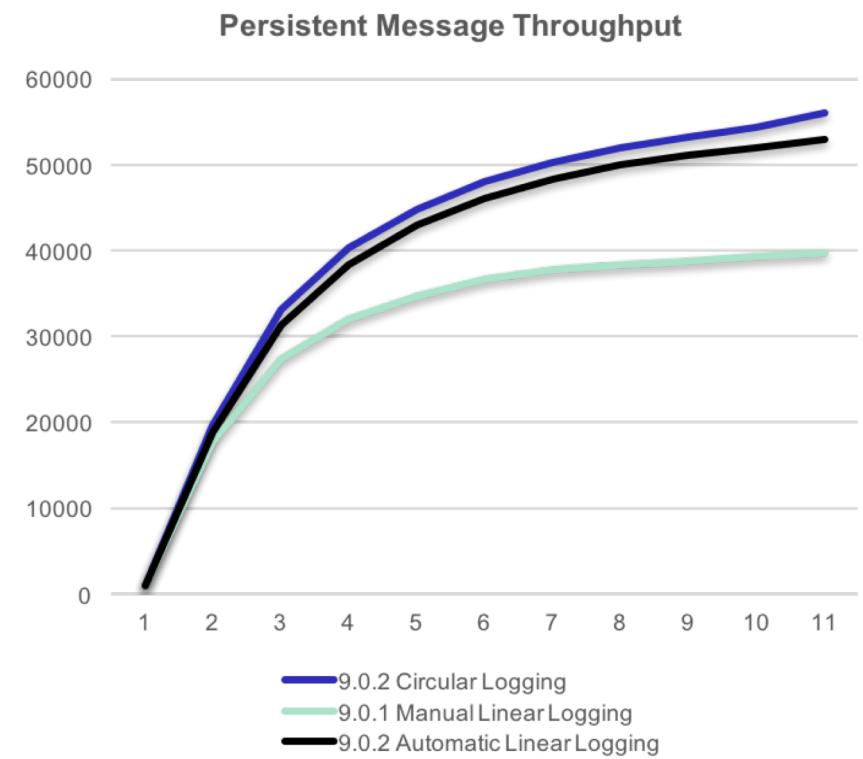
```
SET LOG ARCHIVED(S0000011.LOG)
```

- This indicates that log extent S0000011.LOG is no longer required for archive purposes and once it's no longer required for restart or media recovery it is eligible to be reused
- Logger events are extended to include the oldest extent, ARCHLOG, still needing archiving
- If ARCHLOG gets out of step RESET QMGR TYPE(ARCHLOG) can be used to reset it

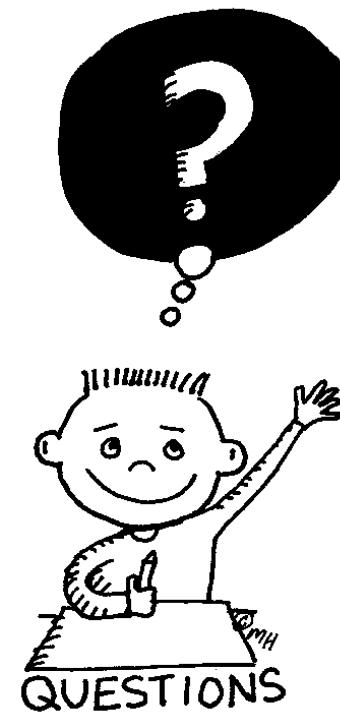
```
-----  
Event Message Received  
Command :Logger Event Command CompCode :0  
Reason :Logger Status  
-----  
Queue Manager Name QM1  
Current Log Extent S0000011.LOG  
Restart Log Extent S0000011.LOG  
Media Log Extent S0000009.LOG  
Log Path /var/mqm/log/QM1/active/  
Archive Log Extent S0000003.LOG  
-----
```

Value of Automatic Reuse of Extents

- Value of reduced administrative burden is obvious
- Value of log reuse shown in this chart:
 - Circular at the top
 - Manual Linear at the bottom
 - Auto Reuse impact on throughput very close to that of Circular



Questions?



Converting Log Files

MIGMQLOG - Why Needed?

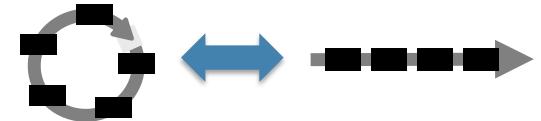
- Wider use of Advanced Format disks posed a problem
 - AF disks use a sector size of 4KB instead of 512 bytes
 - MQ log file pages needed to be reformatted to handle this physical change
- Could have made this change implicitly on first startup following an upgrade
 - But that might pose problems and thus was undesirable
- Could have required queue managers to be deleted and recreated
 - But that would have probably made enemies
- Instead decided to provide a tool to enable customers to do this
 - Wouldn't have to do anything if not needed
 - Could do so at a time of their choosing
- Thus was born the **migmqlog** utility



```
migmqlog -m <QMGR> |-ld <log path> [-align [512|4096]]
```

Since we were in there anyway...

- Made sense to broaden the purpose of this utility
- Make it also possible to change a queue manager's log type
 - Would have the same basic requirement (change the log file page format)
- Command does not change the log data
 - Only modifies log file headers, and renames the log files
 - Also makes appropriate changes to the log control file and qm.ini logging parameters
- Must be run when the queue manager is down
 - Must run successfully to completion for queue manager to be restarted
 - Command is not transactional but is idempotent
 - If the command fails for any reason, simply restart it
- Take care if converting from Linear to Circular and back again
 - Any preexisting media images on the log would be unusable



```
migmqllog -m <QMgr>
           [-ld <new log directory> ]
           [-lc|-ll]
```

Other Changes...

Reducing Log Extents

- The queue manager will attempt to keep as many reusable log extents as needed
 - Goal is to avoid having to create new log extents
 - How many is related to workload as well as how often media images are taken
- If you feel that too much data is being kept, a new command is provided:
 - RESET QMGR TYPE(REDUCELOG)
 - Can be used to remove some or all reusable extents
- This command can also be used for circular logging
 - To release secondary log extents following a large spike in activity

DISPLAY QMSTATUS LOG (1)

- Both runmqsc and PCF now include extra information on DISPLAY QMSTATUS about the state of the log
 - These have been grouped together using DISPLAY QMSTATUS LOG

```
DISPLAY QMSTATUS LOG
  2 : DISPLAY QMSTATUS LOG
AMQ8705: Display Queue Manager Status Details.

  QMNAME (QM1)
  ARCHLOG (S0000011.LOG)
  CURRLOG (S0000018.LOG)
  LOGPATH (/var/mqm/log/QM1/active/)
  LOGUTIL (25)
  MEDIASZ (143)
  RECSZ (1)

  STATUS (RUNNING)
  ARCHSZ (16)
  LOGINUSE (29)
  MEDIALOG (S0000009.LOG)
  RECLOG (S0000018.LOG)
  REUSESZ (49)

  MBs of data available to be reused
  MBs of data needed for restart recovery
  MBs of data just waiting to be archived
  Oldest extent needing archiving
```

The diagram illustrates the relationships between various log file sizes and their usage status. It shows the following components and their connections:

- Log File Sizes:** ARCHLOG (S0000011.LOG), CURRLOG (S0000018.LOG), LOGPATH (/var/mqm/log/QM1/active/), LOGUTIL (25), MEDIASZ (143), RECSZ (1).
- Status and Metrics:** STATUS (RUNNING), ARCHSZ (16), LOGINUSE (29), MEDIALOG (S0000009.LOG), RECLOG (S0000018.LOG), REUSESZ (49).
- Annotations:**
 - Oldest extent needing archiving:** Points to the CURRLOG entry.
 - MBs of data available to be reused:** Points to the REUSESZ metric.
 - MBs of data needed for restart recovery:** Points to the RECSZ metric.
 - MBs of data just waiting to be archived:** Points to the ARCHSZ metric.

DISPLAY QMSTATUS LOG (2)

- Two new statistics are included related to the utilization of the log
- These are useful for both Circular and Linear logging

```
DISPLAY QMSTATUS LOG
```

```
    2 : DISPLAY QMSTATUS LOG
```

```
AMQ8705: Display Queue Manager Status Details.
```

```
    QMNAME (QM1)
```

```
    STATUS (RUNNING)
```

```
    ARCHLOG (S0000011.LOG)
```

```
    ARCHSZ (16)
```

```
    CURRLOG (S0000018.LOG)
```

```
    LOGINUSE (29)
```

```
    LOGPATH (/var/mqm/log/QM1/active/)
```

```
    MEDIALOG (S0000009.LOG)
```

```
    LOGUTIL (25)
```

```
    RECLOG (S0000018.LOG)
```

```
    MEDIASZ (143)
```

```
    REUSESZ (49)
```

```
    RECSZ (1)
```

LOGUTIL is a percentage estimate of how well the queue manager workload is contained within the primary log space

LOGINUSE is the percentage of the primary log space in use for restart recovery at this point in time

Log Record Timestamps

- The *dmpmqlog* tool can be used to print out the contents of the recovery log
- With v9.1, a timestamp has been added to each log record, and will be printed out by the tool
- The timestamp will be in sequence per object, but not necessarily across objects
- Here is an example of the output:

```
LOG RECORD - LSN <0:0:4619:20>
*****
HLG Header: lrecsize 240, version 1, rmid 0, eyecatcher HLRH
Creation Time: 2017-01-21 12:06:45.852 GMT Standard Time (UTC +0)
```

System topics on distributed queue managers

Distributed queue manager information is published to a range of system topic strings

`$SYS/MQ/INFO/QMGR/....`

Authorised subscriptions receive their own stream of publications based on the topic string

Administrative subscriptions

e.g. For information to be continually sent to defined queues

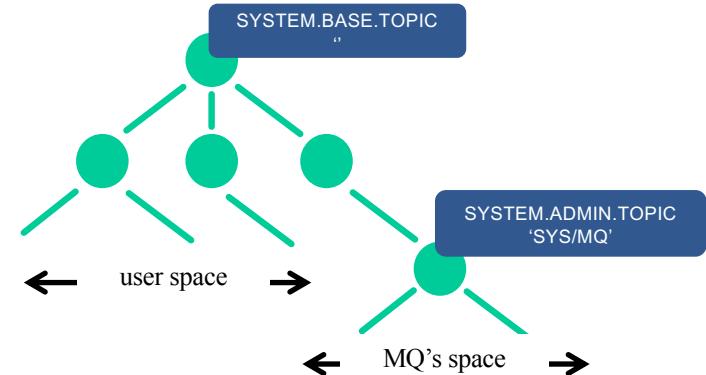
Application subscriptions

e.g. To dynamically listen to information as required

Unlocks system level information for MQ administrators and DevOps teams

Administrators can grant access to subsets of the data,

pertinent to different application teams



Log Monitoring Information

- System topics are new in V9.x and allow you to subscribe to monitoring information periodically generated by the queue manager
- This includes information pertaining to the Recovery Log
 - How efficiently space in the log is being used
 - A measure of the latency the logger is seeing when writing to the log

Log – Physical bytes written

Log – Logical bytes written

Log – Write Latency

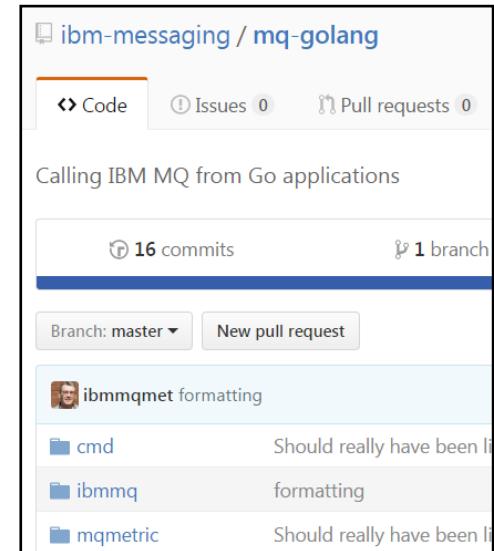


Or use the data to feed your own dashboard
See github.com/ibm-messaging/mq-golang

Looking for How to Do This?

Mark Taylor has an MQ sample available on GitHub

- Written in Go
- Subscribes to MQ9.x \$SYS topics and extracts various MQ and system metrics
- Can be used to push data to various logging servers or databases. MQdev blog entries demonstrate:
 - ▶ Prometheus
 - ▶ Graphic
 - ▶ Logmet
 - ▶ AWS Cloudwatch
 - ▶ InfluxDB
- Could also output to something more generic such as collectd



Monitoring Information

- With v9.1, the Log topic includes information related to the new function introduced originally in v9.0.2:

Log - current primary space in use 29.63%

Log - workload primary space utilization 25.64%

Log - bytes required for media recovery 143MB

Log - bytes occupied by reusable extents 49MB

Log - bytes occupied by extents waiting to be archived 16MB

MQLDT

- Application that simulates the MQ logger
 - ▶ <https://github.com/ibm-messaging/mqldt>
- Blog article
 - ▶ <https://developer.ibm.com/messaging/2017/11/15/mq-log-disk-tester-mqldt/>
- Inputs
 - ▶ Block size
 - ▶ Log configuration
 - ▶ Duration
- Outputs
 - ▶ BW
 - ▶ Latency

Summary

- The Recovery Log a critical component of the queue manager
 - Has been enhanced over time to keep up with modern workloads
- Significant new function added in the v9.1 LTS release related to log management
 - Statistics provide greater insight into logging behavior
 - Automatic Media Imaging can improve performance
 - Automatic Log Management can improve performance and reduce administrative burden

Questions & Answers

