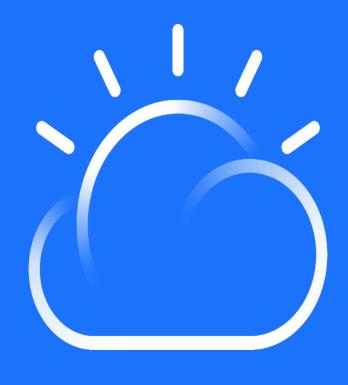
Integration Technical Conference **2019**

M07: Monitoring and problem determination of your MQ distributed systems

Matthew Whitehead mwhitehead@uk.ibm.com







Agenda – the MQ Toolbox

- What's connected?
- Re-routing in a cluster
- Are messages flowing?
- Where are messages going?
- What are the apps doing?
- How can I look back in time?

NB: Capabilities differ in form between distributed and z/OS. This presentation is distributed focused



Presentation contains MQSC example commands and output. Your own admin tools will vary by task, situation and preference. Tasks can be performed in PowerShell, MQ Explorer etc. as desired.



What's connected?



DISPLAY CONN(*) TYPE (HANDLE) ALL

```
AMQ8276: Display Connection details.
                                                                   Use CONN to match TYPE(CONN) and
  CONN (577C425321295301)
                                                                          TYPE(HANDLE) records
  EXTCONN (414D5143474154455741593120202020)
  TYPE (HANDLE)
  OBJNAME (WLMMDB.REQUEST)
                                            OBJTYPE (QUEUE)
  ASTATE (NONE)
                                            HSTATE (INACTIVE)
                                                                    TYPE(HANDLE) records let you find
  OPENOPTS (MQOO OUTPUT, MQOO FAIL IF QUIESCING)
   READA (NO)
                                                                  See all open handles for an app in one place,
  OBJNAME (SENDINGAPP.REPLY)
                                            OBJTYPE (QUEUE)
                                                                         unlike DIS QSTATUS records
  ASTATE (ACTIVE)
                                            HSTATE (ACTIVE)
  OPENOPTS (MQOO INPUT SHARED, MQOO INQUIRE, MQOO SAVE ALL CONTEXT, MQOO FAIL IF QUIESCING)
   READA (NO)
```

DISPLAY CONN(*) ALL

```
AMQ8276: Display Connection details.
   CONN (577C425321295301)
   EXTCONN (414D5143474154455741593120202020)
   TYPE (CONN)
   PID(9740)
                                             TID(185)
  APPLDESC (WebSphere MQ Char
                                            APPLTAG (jms/ATEWAY1 CF)
  APPLTYPE (SYSTEM)
                                             ASTATE (NONE
                                            CONNAME (127.0.0.1)
   CHANNEL (WAS.CLIENTS)
  CONNOPTS (MQCNO SHARED BINDING)
                                            USERID (pbroad)
                                                                     MQ V7.5 and later JMS clients can supply
                                             UOWSTDA (2014-04-08
   UOWLOG()
   UOWSTTI (13.24.00)
                                             UOWLOGDA ( )
                                                                          an application name in the CF
   UOWLOGTI()
                                             URTYPE (XA)
  EXTURID (XA FORMATID [DSAW]
                                                                        Long running UOW information.
XA GTRID[00000145414B8AB4000000104DF48FC00010203040506070809
XA BQUAL[00000145414B8AB4000000104DF48FC00010203040506070809
                                                                  XID can be tied up with app server txn timeout
```

00000000001])



```
DISPLAY CHSTATUS (*) ALL
AMQ8417: Display Channel Status details.
   CHANNEL (WAS.CLIENTS)
                                            CHLTYPE (SVRCONN)
   BUFSRCVD(17)
                                            BUFSSENT (13)
   BYTSRCVD (2296)
                                            BYTSSENT (2456)
   CHSTADA (2014-04-08)
                                            CHSTATI (15.26.59)
   COMPHDR (NONE, NONE)
                                            COMPMSG (NONE, NONE)
  COMPRATE (0,0)
                                            COMPTIME (0,0)
  CONNAME (127.0.0.1)
                                            CURRENT
  EXITTIME (0,0)
                                            HBINT(5)
                                                                   Check suitable heartbeats are negotiated
   JOBNAME (0000260C000000B9)
                                            LOCLADDR ( )
                                            LSTMSGTI (15.26.59)
   LSTMSGDA(2014-04-08)
                                            MCAUSER (pbroad)
  MCASTAT (RUNNING)
  MONCHL (OFF)
                                            MSGS(6)
                                            SSLCERTI (CN=ExampleCA, O=Example)
  RAPPLTAG(jar)
                                            SSLKEYTI()
   SSLKEYDA ( )
  SSLPEER (SERIALNUMBER= i3:43:FD:D6, CN=ExampleApp1, O=Example)
                                            STATUS (RUNNING)
   SSLRKEYS (0)
   STOPREO (NO)
                                            SUBSTATE (RECEIVE)
   CURSHCNV (1)
                                            MAXSHCNV (1)
   RVERSION ( 0000000)
                                            RPRODUCT (MQJM)
                                        See SSLPEER information not in DIS CONN
                      JOBNAME contains PID (except z/OS): 0x260C = PID(9740)
                      On Linux/UNIX (not Win) TID matches CONN: 0xB9 = TID(185)
```

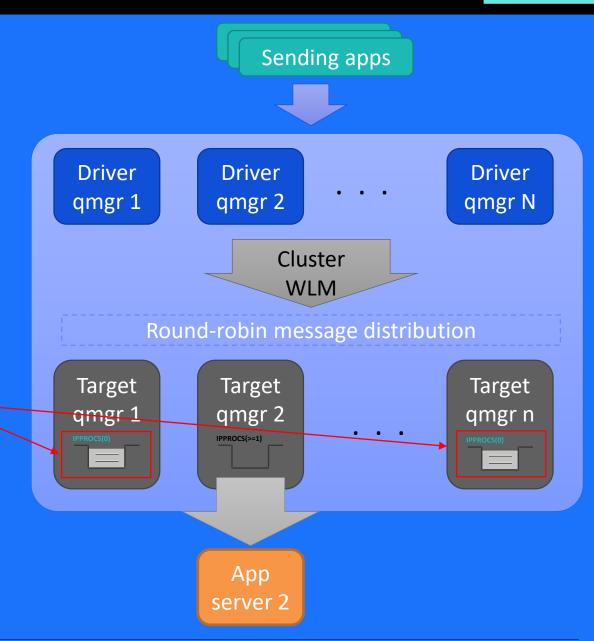
Note that multiple CONN might share one SVRCONN channel instance



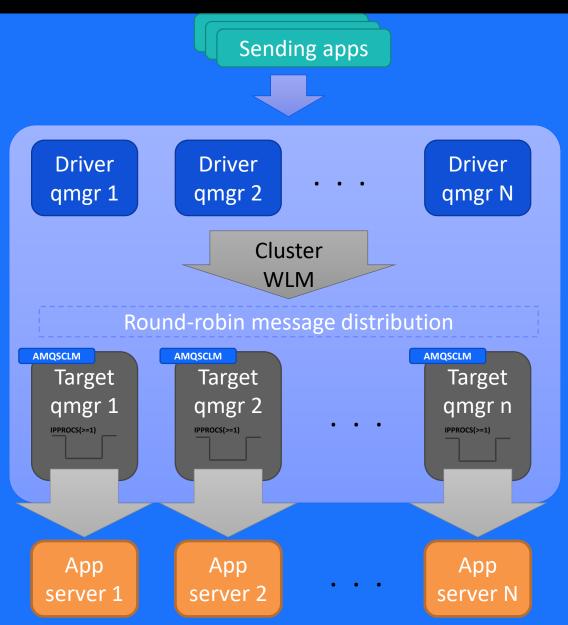
Rerouting messages if nobody is connected

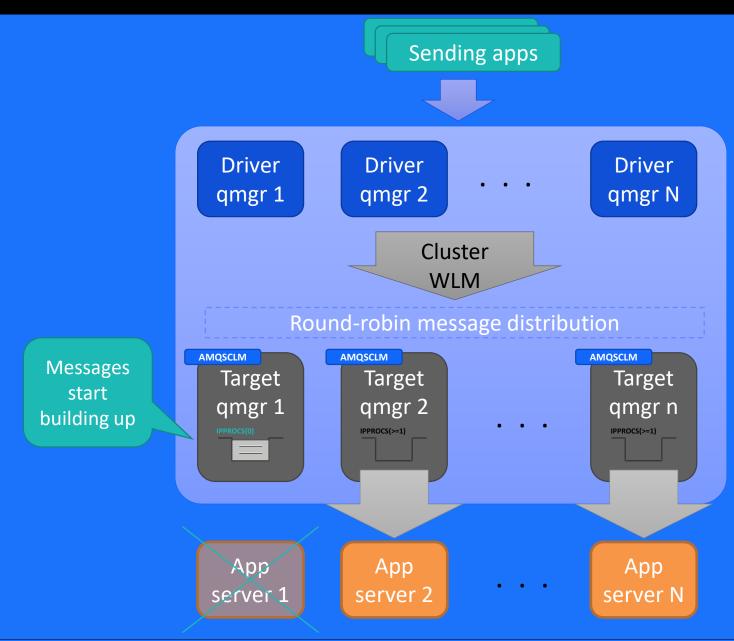


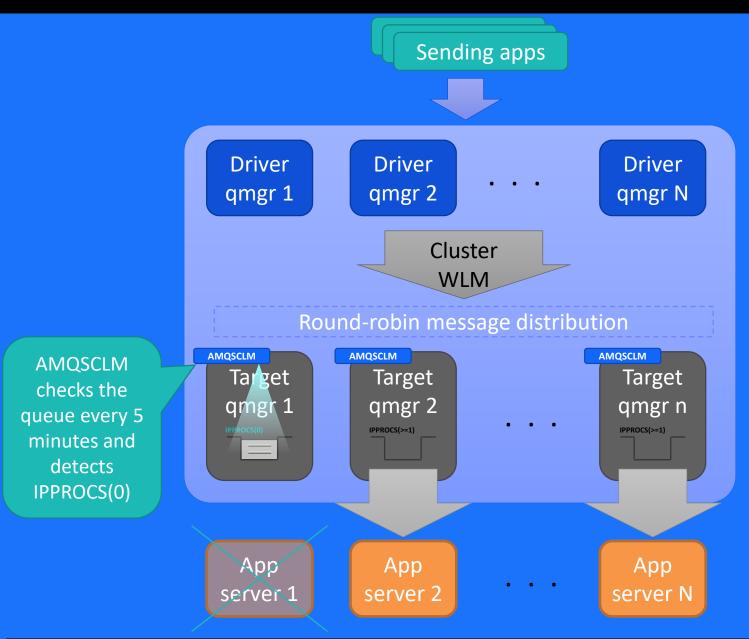
- Cluster WLM algorithm doesn't check whether consuming applications are connected to cluster queues
 - i.e. whether something is getting messages from them
- You need to ensure that applications are consuming from all instances of a clustered queue to prevent messages building up
- However, there is an alternative: AMQSCLM

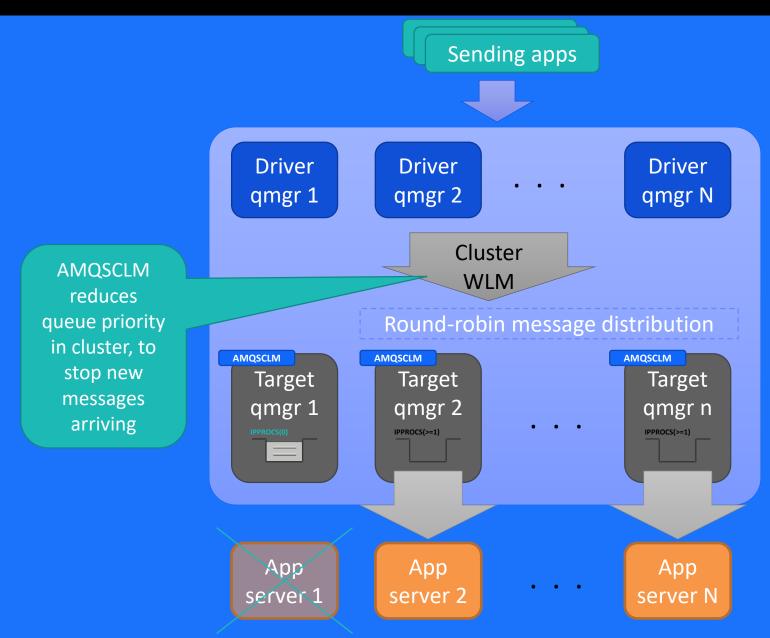


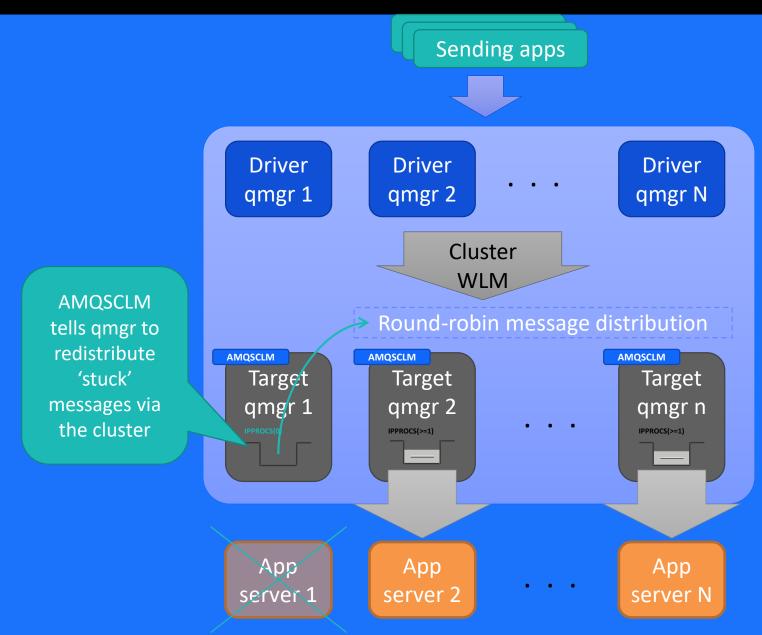












AMQSCLM summary

- All platforms

- The cluster queue monitoring sample program (AMQSCLM)
- Shipped with the product as a sample
 - Precompiled
 - Source code
 - Not shipped on z/OS, but distributed source code can be compiled and used on z/OS
- More information in the MQ Knowledge Center



Are messages flowing?



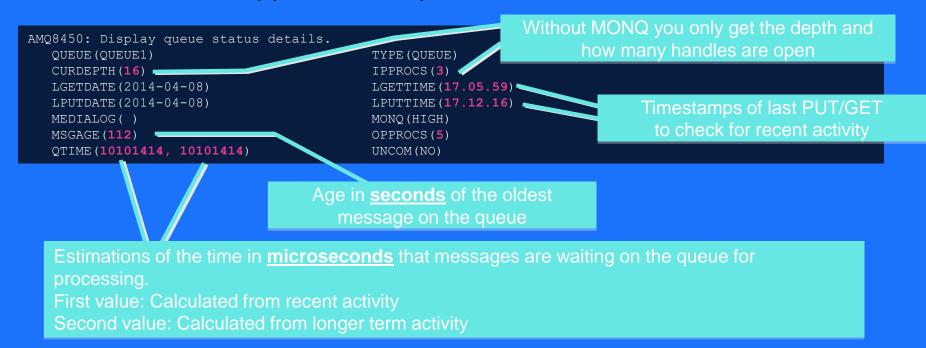


- Set detail level for queue manager. Override for individual queues
 - LOW, MEDIUM and HIGH set frequency of sampling, not type of stats generated

```
ALTER QMGR MONQ (MEDIUM)
ALTER QLOCAL (QUEUE1) MONQ (HIGH)
ALTER QLOCAL (QUEUE2) MONQ (OFF)

DIS QSTATUS (QUEUE1) ALL
```

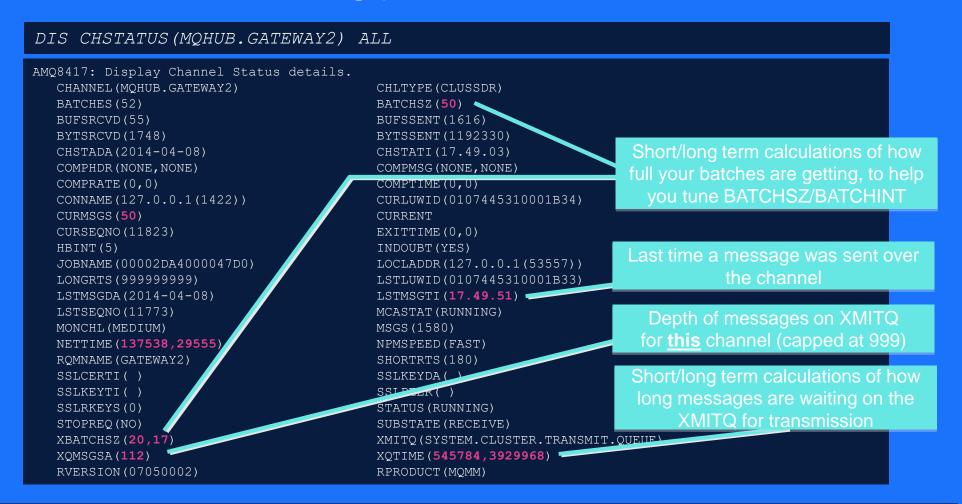
Gives live view of application responsiveness





```
ALTER QMGR MONCHL (MEDIUM) MONACLS (MEDIUM)
ALTER CHANNEL (CLUSTER1.QM1) CHLTYPE (CLUSRCVR) MONCHL (HIGH)
```

Gives live view of channel throughput



New options in

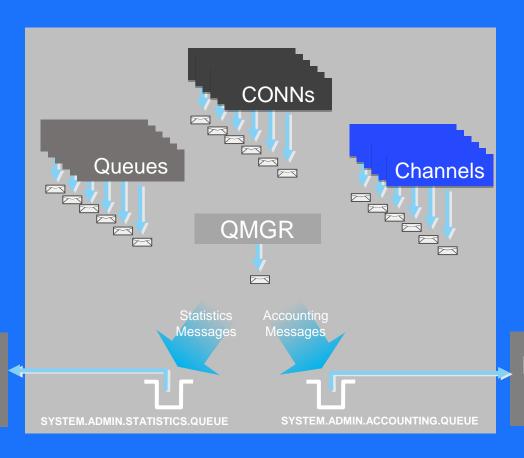
Use SMF on z/OS



Related attributes

- STATQ
- STATMQI
- STATCHL
- **STATACLS**
- STATINT

PCF statistics collection app



Related attributes

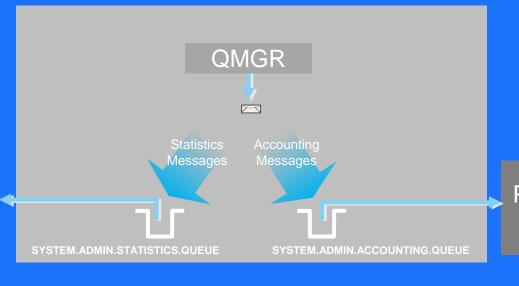
- ACCTQ
- **ACCTMQI**
- ACCTINT

PCF accounting collection app

- Monitoring data sent as a PCF message at a configured interval
 - Statistics scoped to a queue / channel / QMGR
 - Accounting scoped to an individual CONN and queue / QMGR

New options in V9

Use SMF on z/OS



PCF accounting collection app

e.g.

MSOP Explorer Plugin

PCF statistics

collection app

(SupportPac)

amqsmon

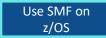
Specifically for accounting and statistics

amgsevt

Generic PCF event message formatter, TEXT or JSON output Other

- Tivoli
- Custom app
- E.g. Java app using com.ibm.mq.headers

Taking a look with SupportPac MSOP





Queue Manager: GATEWAY1

Last Operation: Reading from SYSTEM.ADMIN.STATISTICS.QUEUE

i Mot showing TDO details

S From 2014-04-08 23.44.55 to 2014-04-08 23.48.55

Disconnections : 236

Disconnects : 215

		•	
Verb		Success	Fail
0pen	:	19187	0
Close	:	19170	0
Inq	:	2544	1
Set	:	1	0

- Actions on Queue Manager
- 👺 Other Actions
- Messages
- ⊳ 🖹 Publish/Subscribe

▶ Queue Name : SYSTEM.ADMIN.COMMAND.QUEUE

D Queue Name : SYSTEM.CLUSTER.COMMAND.QUEUE

▶ Queue Name : SYSTEM.PROTECTION.POLICY.QUEUE

🔈 🔼 Queue Name : WLMMDB.BACKOUT

🗸 🖂 Queue Name : WLMMDB.REQUEST

Created : 2014-01-22 15.33.19

Queue Type : Local

Def Type : Predefined

Max Q Depth : 462

Min Q Depth : 0

Message Type		Non-persistent	Persistent
Put	:	0	2779
Putl	:	0	0
Get	:	0	2319
Browse	:	0	0
Put Bytes	:	0	772345
Get Bytes	:	0	644502
Browse Bytes	:	0	0
Average Life		Π	11358845

Time period

MQ Statistics at QMGR level

Detailed queue statistics

Taking a look at <u>accounting</u> events with amqsevt

```
mwhitehead@ubuntu: ~
File Edit View Search Terminal Help
mwhitehead@ubuntu:~$ /opt/mqm/samp/bin/amqsevt -m QMCONF -q SYSTEM.ADMIN.ACCOUNTING.QUEUE -o json
                  "objectName": "SYSTEM.ADMIN.ACCOUNTING.QUEUE",
                    "objectTvpe" : "Queue" },
"eventType" : {
   "name" : "Accounting MQI"
    value : 10/
'eventReason" : {
   "name" : "None".
   "value" : 0
'eventCreation" : {
   "timeStamp" : "2018-09-13T08:13:542",
                : 1536826434
   "epoch"
'eventData" : {
 "queueMgrName" : "OMCONF"
 "startDate" : "2018-09-13"
 "startTime" : "01.13.54",
 "endDate": "2018-09-13",
 "endTime" : "01.13.54",
  "commandLevel" : 910,
 "connectionId" : "414D5143514D434F4E46202020202020172.995B109B8423",
  'applName" : "amqsbcg"
  "processId" : 65110,
 "threadId" : 1,
 "userIdentifier" : "mwhitehead",
 "connDate" : "2018-09-13",
 "connTime" : '01.13.54",
 "discDate" : 2018-09-13"
```

Choose JSON or TEXT format

Which application does this event relate to?

Which type of event is this: ACCTMQI or ACCTQ?

Taking a look at <u>statistics</u> events with amqsevt

```
mwhitehead@ubuntu: ~
File Edit View Search Terminal Help
mwhitehead@ubuntu:~$ /opt/mqm/samp/bin/amqsevt -m QMCONF -q SYSTEM.ADMIN.STATISTICS.QUEUE -o json
"objectType" : "Queue" },
"eventType" : {
   "name" : "Statistics MQI"
    'value" : 164
eventReason" : {
   "name" : "None",
   "value" : 0
'eventCreation" : {
   "timeStamp" : "2018-09-13T09:09:152"
   "epoch"
               : 1536829755
'eventData" : {
  "queuelly Name" . "QHCONF",
  "startDate" : "2018-09-13"
  "startTime" : "02.08.39",
  "endDate" : "2018-09-13",
  "endTime" : "02.09.15",
   commandLevel" . 010,
  "conns" : 1,
 "connsFailed" : 0,
 "connsMax" : 23,
 "discs" : [
   1,
```

Choose JSON or TEXT format

Period of time the statistics relate to

Which type of event is this: STATMQ, STATQ, STATCHL, or STATACLS?

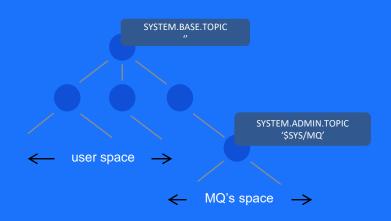
Use SMF on z/OS

```
ALTER QMGR STATMQI(ON)
Wait a bit, but not the default 30 minutes between stats records
RESET QMGR TYPE(STATISTICS)
amqsmon -m GATEWAY1 -t statistics -a -w 0
```

```
MonitoringType: MQIStatistics
OueueManager: 'GATEWAY1'
IntervalStartDate: '2014-04-09'
IntervalStartTime: '00.00.35'
IntervalEndDate: '2014-04-09'
IntervalEndTime: '00.01.13'
CommandLevel: 700
ConnCount: 35
PutCount: [271, 0]
PutFailCount: 0
Put1Count: [2, 0]
Put1FailCount: 0
PutBytes: [273976, 0]
GetCount: [270, 0]
GetBytes: [269468, 0]
GetFailCount: 19
DurableSubscriptionHighWater: [0, 0, 0, 0]
DurableSubscriptionLowWater: [0, 0, 0, 0]
NonDurableSubscriptionHighWater: [0, 0, 0, 0]
NonDurableSubscriptionLowWater: [0, 0, 0, 0]
PutTopicCount: [0, 0]
PutTopicFailCount: 0
Put1TopicCount: [0, 0]
Put1TopicFailCount: 0
PutTopicBytes: [0, 0]
PublishMsqCount: [0, 0]
PublishMsgBytes: [0, 0]
```

- Overall QMGR busyness
- Simple data format
 - Multiple values are [Persistent, NonPersistent]
- One message every X seconds
 - Use amqsmon directly (perl/cron)
- Low/high water marks for subscriptions
 - Grouped by subscription type
- amqsmon is a sample so you can use it as a base for your own tools

- 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



- Familiar statistics available through subscriptions
 - Queue manager wide statistics (connects, disconnects, opens, closes, puts, gets, ...)
 - Queue level statistics (opens, closes, puts, gets, ...)
 - NB: statistics available from system topics are not a 1-1 mapping to those available from system queues
 - E.g no channel statistics, some missing information, some new information, some merged information
 - No support for accounting data
- Extended to include CPU and disk usage. For example...
 - Queue manager CPU time, memory usage
 - Disk reads/writes, disk latency,
- Subscribe to meta-topic to learn which classes of statistics are available
 - \$SYS/MQ/INFO/QMGR/QMGR1/Monitor/METADATA/CLASSES
 - Then subscribe to specific topics
 - \$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
 - See amqsrua sample program

- By default messages are published every 10 seconds
 - Configurable via a tuning parameter in qm.ini, e.g.

TuningParameters:

MonitorPublishHeartBeat=60

 Recent updates to amqsrua sample program let you subscribe to other parts of the \$SYS topic tree



MQ V9.0.2

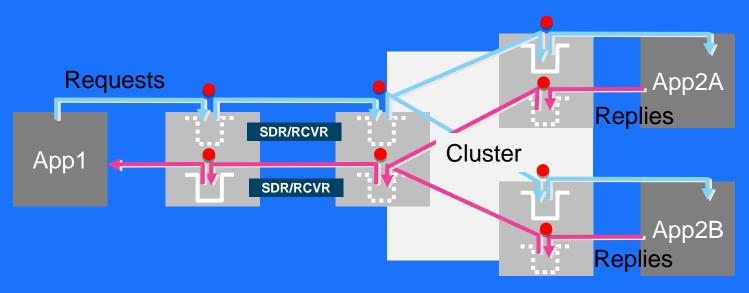
- amqsrua –p "\\$SYS/Application/runmqsfb"
- Applications can publish their own meta-data and events to \$SYS/Application
- Single place to consume MQ and application events from



```
$ amgsrua -m V9000 A
CPU: Platform central processing units
DISK: Platform persistent data stores
STATMOI : API usage statistics
STATQ: API per-queue usage statistics
Enter Class selection
==> CPU
SystemSummary: CPU performance - platform wide
QMgrSummary: CPU performance - running queue manager
Enter Type selection
==> SystemSummary
Publication received PutDate: 20160411 PutTime: 10465573
User CPU time percentage 0.01%
System CPU time percentage 1.30%
CPU load - one minute average 8.00
CPU load - five minute average 7.50
CPU load - fifteen minute average 7.30
RAM free percentage 2.02%
RAM total bytes 8192MB
Publication received PutDate: 20160411 PutTime: 10466573
User CPU time percentage 0.01%
System CPU time percentage 1.30%
```

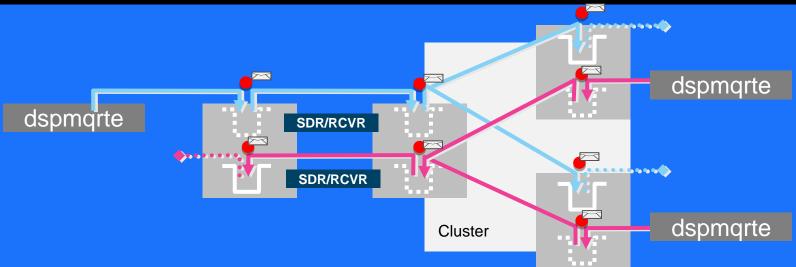


Where are messages going?



- At each of the dots stuck / mis-sent messages are possible
 - MQOPENs of the wrong queue / queue manager by apps
 - Full queues
 - Stopped channels
 - Stopped apps
 - Incorrectly configured QREMOTE/QALIAS routing objects
 - Cluster membership problems
- The standard problem diagnosis approach
 - Methodically checking channels/queues/DLQs at each point
- Is there anything to speed up this process?





- MQ has the ability to inject trace route messages
 - (Can be) hidden from applications
 - Generate activity reports as they pass through, potentially accumulated in the message
- Tools are available to trace routes using these reports
 - dspmqrte command line tool supplied with the product
 - MSOP Cat 2 SupportPac extension to MQ Explorer
- Lets you see the path messages could have taken
 - Test connectivity through the MQ network
 - · Test cluster workload balancing
- Can quickly jump you close to the problem
 - The point your trace message veers off in the wrong direction
 - · The point the trail goes cold
- NB: there is also a related technology: activity recording which generates activity reports from **real** messages



What are the apps doing?

- Information on all the MQI operations performed by an application in the order that they were done
- Similar infrastructure to accounting & statistics
 - PCF messages on SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE
 - Configurable via mqat.ini
 - Can be changed without queue manager restart
 - Configurable detail level can include partial/full message payload
 - Frequency options for tuning
 - Can be enabled on a per-application basis
 - Via MQCONNX flags
 - Via application name

Health warning: Performance impact

http://ow.lv/vA8wB





Enables scenarios such as

Not on z/OS

- Application audit trail
- Message duplication
- Resource usage
 - Which queues or topics are actually being used
- Problem determination
 - Which queue / queue manager is the application actually opening
- Application coding standards
 - Does everyone use the MQI in the recommended way
- And more ...

Health warning: Performance impact





Looking at the data with the amqsact sample |

Put_time: '00394866'conference 2019

```
ALTER QMGR ACTVTRC (ON)
                                                           should be tuned via mgat.ini
amgsact -m GATEWAY1
MQI Operation: 6
  Operation Id: MQXF PUT
 ApplicationTid: 12451
  OperationDate: '2014-04-09'
 OperationTime: '01:39:48'
  High Res Time: 1397003988665548
  Completion Code: MQCC OK
                                                                                               everything the QM has generated (-v)
 Reason Code: 0
  Hobj: 18225032
  Put Options: 139330
 Msg length: 460
  Recs present: 0
  Known dest count: 1
                                                                                 Check the options used for coding standards
  Unknown dest count: 0
  Invalid dest count: 0
  Object type: MQOT Q
 Object name: 'SENDINGAPP.REPLY'
  Object Q mgr name: 'GATEWAY1'
  Resolved Q Name: 'SENDINGAPP.REPLY'
 Resolved Q mgr: 'GATEWAY1'
                                                                                Check queue name resolution, to find out
  Resolved local Q name: 'SENDINGAPP.REPLY'
  Resolved local Q mgr: 'GATEWAY1'
                                                                              why messages are going to the wrong place
  Resolved type: MQOT Q
  Report Options: 0
 Msg type: MQMT DATAGRAM
  Expiry: -1
  Format name: 'MQHRF2'
  Persistence: 0
  Msq id:
  00000000: 414D 5120 4741 5445 5741 5931 2020 2020 'AMO GATEWAY1
  00000010: 0207 4453 2007 2603
                                                  '..DS .&.
  Correl id:
  00000000: 414D 5120 4741 5445 5741 5931 2020 2020
                                                  'AMO GATEWAY1
  00000010: 0207 4453 2007 2203
                                                  '..DS .".
  Reply to Q : '
 Reply to Q Mgr: '
  Coded char set id: 1208
  Encoding: 273
  Put date: '20140409'
```

- Application activity trace enabled through subscriptions rather than queue manager configuration
- Subscribe to topic
 - E.g. \$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ApplName/amqsput
 - Filter by application name, channel or connection id
- When a subscription is created, PCF messages start to flow to the subscriber's queue. When subscription is deleted, messages stop
- Much easier to get just the data you want!

- Sample provided to demonstrate usage and format output
- Example below specifies application name (-a) so uses dynamic mode
- Dynamic mode subscribes to system topic rather than uses system queue
- Channel name and connection id also supported

```
$ amgsact -m QMGR1 -a amgsput -w 60
Subscribing to the activity trace topic:
  '$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ApplName/amgsput'
MonitoringType: MQI Activity Trace
                                                                             $
QueueManager: 'QMGR1'
ApplicationName: 'amgsput'
Application Type: MQAT UNIX
  Tid Date
                 Time
                            Operation
                                            CompCode
                                                          MQRC
                                                                HObj (ObjName)
  001 2016-04-14 09:56:53
                           MOXF CONNX
                                            MQCC OK
                                                           0000
                           MOXF OPEN
                                            MQCC OK
                                                                 2 (QUEUE1)
  001 2016-04-14 09:56:53
  001 2016-04-14 09:56:53
                           MQXF PUT
                                            MQCC OK
                                                           0000
                                                                 2 (QUEUE1)
                           MOXF PUT
                                                                 2 (QUEUE1)
  001 2016-04-14 09:56:53
                                            MQCC OK
                                                           0000
                                                                 2 (QUEUE1)
  001 2016-04-14 09:56:53
                           MQXF CLOSE
                                            MQCC OK
                                                           0000
  001 2016-04-14 09:56:53
                            MQXF DISC
                                            MQCC OK
                                                           0000
```

```
$ amqsput QUEUE1 QMGR1
Sample AMQSPUT0 start
target queue is QUEUE1
Hello
World
Sample AMQSPUT0 end
$
```



How can I look back in time?

What happened to my messages at 2am this morning?

- Enterprise monitoring solution
 - DLQ alerts, queue depth alerts, channel status alerts
 - Unresolved running units of work
 - Historical MQ monitoring, accounting and stats data
- App logs from the time of the problem
 - Exceptions, MQ error codes, timeouts
- MQ error logs for all qmgrs that could have been involved
 - Channel errors
 - Authentication issues
- ??? what else is there



- For persistent messages inside transactions
 - MQ logs each operation performed
 - Outside of transactions persistent messages might be logged
- Why can't we use this to
 - Look back in time to 2am and see what happened?
 - Recover the original payload if the app lost the message?
 - See what happened inside long-running units of work?
 - Provide a list of operations within the failed business transaction?
- MQ documents how you can... if
 - You use the text formatting tool provided with MQ (dmpmqlog)
 - The logging is linear so the historical data is available in the tool
 - You follow the right steps to extract data from running qmgrs
 - You do the work to follow through the logs

dmpmqlog output is readable, but analysis is tedious

```
LOG RECORD - LSN <0:0:954:44817>
*****
HLG Header: lrecsize 873, version 1, rmid _____eatcher HLRH
LogRecdType . . : AQM Put Message (257)
LogRecdLen . . : 853
QueueName . . . : Not known
Qid . . . . . : {Hash 2147211283, Counter: 5}
ThisLSN . . . : <0:0:0:0>
Version . . . . : 4
MapIndex . . . : 199
                                PrevLink.Length: 8
PrevDataLink . : {High 0, Low 103424}
Data.Locn . . : 103424
                                Data.Length . . : 613
00000: 41 51 52 48 04 00 00 00 FF FF FF FF FF FF FF FF
                                                 AQRH....ÿÿÿÿÿÿÿÿ
00016: 00 00 00 00 00 00 00 c7 00 00 00 02 00 c0 01
00032: 00 00 00 04 00 01 00 A5 00 00 00 00 00 00
                                                 c...AMQ IIB01 QM
      20 20 20 20 D9 26 B0 52 20 08 53 F5 30 30 30 30
      30 30 39 39 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 FF FF FF FF 00 00 00 04 00 00 09
      00 00 00 00 E3 ED 04 80 10 FD 67 E4 FF FF FF
                                                 ....ãí.€.ýgäÿÿÿÿ
      4D 44 20 20 01 00 00 00 00 00 00 00 08 00 00 00
      00 00 00 00 11 01 00 00 B8 04 00 00 4D 51 48 52
      46 32 20 20 04 00 00 00 01 00 00 00 20 20 20 20
      20 20 20 20 20 20 20 20 20 20 20 20 49 49 42 30
      31 5F 51 4D 20 20 20 20 20 20 20 20 20 20 20 20 20
                                                 1 QM
      20 20 20 20 20 20 20 20 20 20 20 20 4D 55 53 52
                                                           MUSR
      5F 4D 51 41 44 4D 49 4E 16 01 05 15 00 00 00 64
                                                 MQADMIN.....d
      20 3E AC 57 48 B3 09 B8 71 B0 4C F2 03 00 00 00
     00 00 00 00 00 00 00 0B 20 20 20 20 20 20 20 20
      20 20 20 20 20 20 20 20 10 00 00 00 57 65 62 53
                                                        ....WebS
      70 68 65 72 65 20 4D 51 20 43 6C 69 65 6E 74 20
                                                 phere MQ Client
      66 6F 72 20 4A 61 76 61 32 30 31 33 31 32 31 39
      31 34 32 32 33 32 32 32 20 20 20 20 00 00 00 00
      ..... <mcd><Ms
     64 3E 6A 6D 73 5F 74 65 78 74 3C 2F 4D 73 64 3E
      3C 2F 6D 63 64 3E 20 20 00 00 00 58 3C 6A 6D 73
                                                 </mcd> ...X<jms
      3E 3C 44 73 74 3E 71 75 65 75 65 3A 2F 2F 2F 51
      31 3C 2F 44 73 74 3E 3C 54 6D 73 3E 31 33 38 37
                                                 1</Dst><Tms>1387
                                                 462952226</Tms><
      34 36 32 39 35 32 32 32 36 3C 2F 54 6D 73 3E 3C
      43 69 64 3E 30 30 30 30 30 39 39 3C 2F 43 69
      64 3E 3C 44 6C 76 3E 32 3C 2F 44 6C 76 3E 3C 2F
                                                 d><D1v>2</D1v></
      6A 6D 73 3E 61
```

Ordered unique IDs for each record (LSN)

A set of documented record types

Transaction information with XIDs, or re-used MQ transaction IDs

MQMD header data at discoverable offsets in the hex of a message Put

The message payload itself

 Wouldn't it be easier to let the computer do the tedious bit?



- Takes the tedium out of analysing the output from dmpmqlog
- Created by Peter Broadhurst
- Download from http://www.ibm.com/support/docview.wss?uid=swg21660642

```
java -jar dmpmqlog.scraper-20151201.jar -b little-endian -i dmpmqlog.txt -o .
```

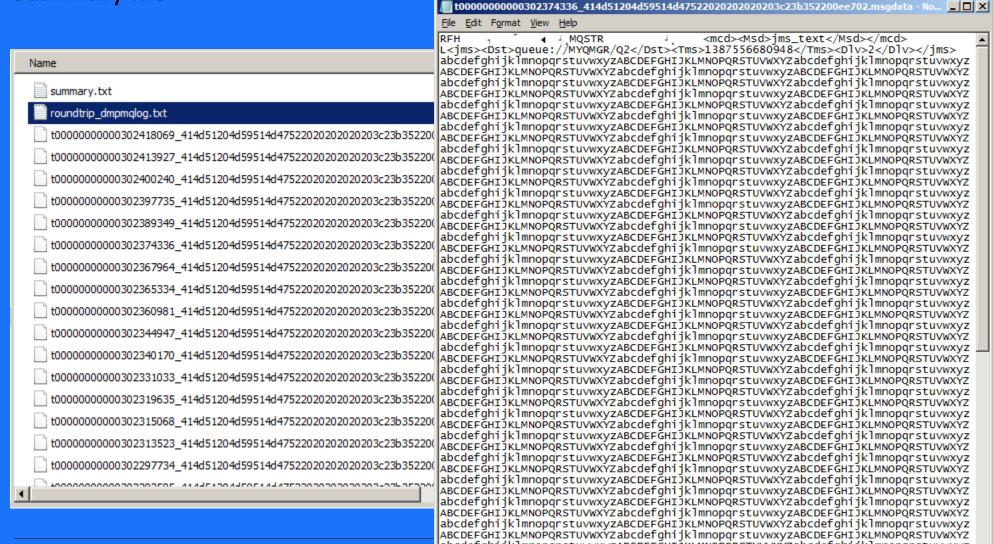
- Generates file per message PUT in the supplied data
- Summary file

dmpmqlog scraper tool output



Generates file per message PUT in the supplied data.

Summary file



<mcd><msd>jms_text</msd></mcd> L<jms><Dst>queue://MYQMGR/Q2</Dst><Tms>1387556680948</Tms><Dlv>2</Dlv></jms> abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefqhijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopgrstuvwxyz |ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefqhijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz ABCDEFĞHIJKLMNOPQRSTUVWXYZabcdefqhijklmnopqrstuvwxyzABCDEFĞHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopgrstuvwxyz



- Lots of tools in your MQ toolbox!
- On-line status commands
 - DISPLAY CONN
 - DISPLAY QSTATUS
 - DISPLAY CHSTATUS
- Cluster monitoring AMQSCLM
- Off-line statistics and accounting
 - amqsmon and MSOP to view
- Tracking
 - Trace-route
 - Application activity trace
- MQ recovery logs
 - dmpmqlog scraper



Thank You

