Analysing MQ SMF Data

Mark Taylor, IBM Hursley
marke_taylor@uk.ibm.com

Lyn Elkins, IBM WSC
elkinsc@us.ibm.com

What is SMF

- Not the Sacramento Music Festival
- SMF is the z/OS System Management Facility
  - A common utility for all z/OS subsystems to report activity
  - What they report and when they report it is up to them
- Each subsystem is assigned one or more SMF Types
  - MQ uses:
    - SMF 115 – or MQ Statistics records
    - SMF 116 – or MQ Accounting records
- SMF data is (1) collected, (2) dumped to a data set, then (3) **formatted and analysed**

How do you collect data (1)

- Collecting MQ SMF is controlled two ways:
- SYSP Macro
  - SMFSTAT attribute owns the Statistical (SMF 115) record production
    - This should be on all the time, set to (01,04) for all queue managers
  - SMFACCT attribute owns the Accounting (SMF 116) record production
    - Typically not on all the time, controlled by the START TRACE command

How do you collect data (2)

- Collecting MQ SMF is controlled two ways:
- START TRACE command
  - +cpf START TRACE(A) CLASS(?)
- Starting and stopping the accounting trace is typically dynamic
  - CLASS(3) controls production of the Task Accounting (including queue) data
  - CLASS(4) controls production of the Channel Accounting data
  - CLASS(1) is no longer used
  - The classes are not inclusive, so if you want both Task and Channel accounting you need to turn on both 3 and 4

Dumping records to SMF output data set

- Once SMF data has been collected, it must be sent to output data sets
  - These can be considered an intermediate state
  - IFASMFDP copies data from SMF data sets to a sequential data set
  - IFASMFDL copies data from SMF logstreams to a sequential data set
- Options filter which records are copied to the output data set
  - Once the output data set is created, then it can be formatted

SMF Streaming

- New capability with z/OS 2.x via PTF OA49263
- “Live” access to SMF buffers without needing to dump for offline processing
  - Can then process data for real-time analytics
- Tool described here does not exploit that. But the SQL examples could be used

Working with the data

- Various tools have been around for a while
- CSQ4SMFD is a sample program provided with MQ
  - Dumps records from the data sets created by IFASMFDP/L jobs in a readable but unconsumable format
- SupportPac MP1B – free tool to create reports from records
- Other commercially available tools for interpretation
  - Sometimes do not keep up with changes
  - Do not capture/use some critical data
  - This column means what?

CSQ4SMFD Output – Message Manager

```
 message manager statistics data
--Q-M-S-T---H-E-X---P-R-I-N-T----
Address   = 13B2AC08
00000000 : D40F0048 D8D4E2E3 00000001 00000001 <M...QMST........>
00000010 : 00000013 00000003 00000000 00000002 <................>
00000020 : 00000000 00000000 00000000 00000000 <................>
00000030 : 00000000 00000000 00000001 00000008 <................>
00000040 : 00000000 00000000                   <........        >
--Q-M-S-T---F-O-R-M-A-T-T-E-D----
qmstid   = d40f
qmstll   = 0072
qmsteyec = QMST
qmstopen = 00000001
qmstclos = 00000001
qmstget  = 00000019
qmstput  = 00000003
qmstput1 = 00000000
qmstinq  = 00000002
```

Challenges

- Tools sometimes broke with different MQ levels
- Calculations were not always clear, or correct
- Difficult to validate they were doing the right thing
- Filled up JES spool with reports

JES Spool Example

SMF Record Type	Number of records
2	1
3	1
115	66
116	2,684,149

$HASP375 ELKINSE1 ESTIMATE EXCEEDED BY 167,100,000

Output file name	Number of Lines
BUFF	316
BUFFCSV	57
LOG	326
TASK	163M

Challenges

- Would get calls asking how formatters actually worked
  - As I could see source code
- Not always able to understand it
  - But could see inconsistencies

Solution

- I decided I had to learn how to process SMF

- Investigation ...

- Found various tools and toolkits but none suitable
  - Java code that only runs on z/OS because of I/O
  - Parser using DFDL for IIB records

As a Distributed person

- I know how to develop code that runs on Unix and Windows
  - Editors, compilers, debuggers etc

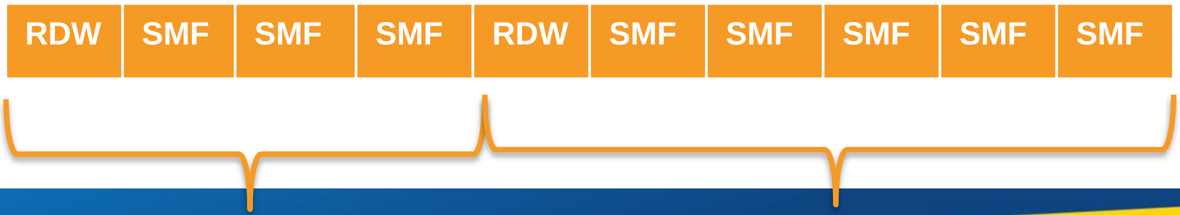
- Lyn wanted to import to spreadsheets which run on those platforms
  - And different programs were better able to handle large data
  - So formatting SMF on these platforms made sense

Project Goal

- Develop a tool that did not get in the way of analyses
- Format all the data and nothing but the data
- Syntax. Not semantics.

Some issues

- Formatting RDW
  - z/OS data sets are structured (embedded record lengths)
  - Files on Unix/Windows are mostly byte-streams
  - Need to be able to deal with the Record Descriptor Words
    - ftp options to keep RDW bytes when transferring bytes
    - > QUOTE SITE RDW
    - > BINARY



Yet more issues

- C headers and Assembler macros did not always match
- Incompatible changes made across some versions
  - Fields inserted in middle of structures
- Data formats not always cross-platform C-friendly
  - Assumptions about data type sizes
  - Assumptions about bit fields
  - Assumptions about endian-ness
  - Assumptions about padding
  - Structures not always complete/overlap

SMF not as self-describing as advertised

- Despite claims, SMF is not really self-describing
  – Unlike PCF
  – Model is header followed by “triplets” which say where each real element is, how long it is, and how many there are
- MQ SMF has some undocumented triplets, or skipped fields
  – Can’t tell without reading docs and looking at the sample source code (and sometimes verifying in product source code)
  – Not everything has an eyecatcher (newest CHIN records)

Starting on the formatter

- Started with RDW record reader, hex and EBCDIC dumper
  - Similar to the raw output from CSQ4SMFD
- To ensure I was processing one complete record at a time
  - One SMF record may be split across multiple dataset records
- Program evolved …
  - Simple structure for formatting MQ structures such as QPST
  - Adding the V9 pageset statistics took minutes
  - Might choose a different approach (Java?) if restarting
- Adding other record types (AMS is 180) is feasible

Post Processing Challenges

- Formatting the output data also had “opportunities”

- Spreadsheets try to be clever when importing CSVs
  - Date, time formats
  - Treating strings as numbers
  - And sometimes get it wrong
- So this formatter went through several iterations testing with Excel and LibreOffice to ensure data could be imported
  - Compromises needed on timestamp formats

Unexpected popularity

- After first version running, mentioned it at Interconnect 2016
- "How many people interested"
  - Expected only the co-presenter to raise hand
  - Rather more than that did
- So quickly got a version on github

Github project

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

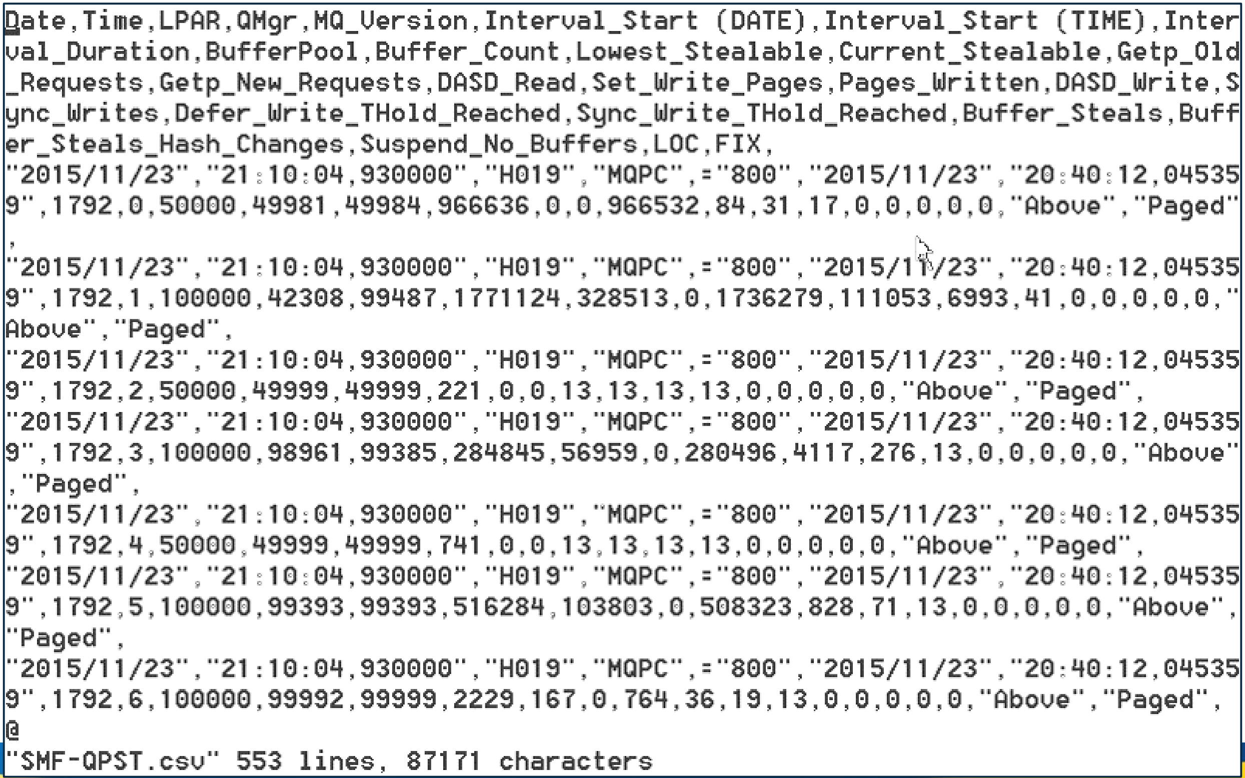
Downloading data

```
$ ftp winmvs41
Connected to 9.20.1.1
User (winmvs41:(none)): met
331 Send password please.
Password:
230 MET is logged on. Working directory is “MET.”.
ftp> BINARY
200 Representation type is Image
ftp> QUOTE SITE RDW
200 SITE command was accepted
ftp> GET ‘MET.SMF.DATA’ c:\smf\data\test.bin
200 Port request OK
125 Sending data set MET.SMF.DATA
250 Transfer completed successfully
ftp: 792532 bytes received in 0.30 Seconds 2641.77Kbytes/sec.
ftp> quit
```

Running the program

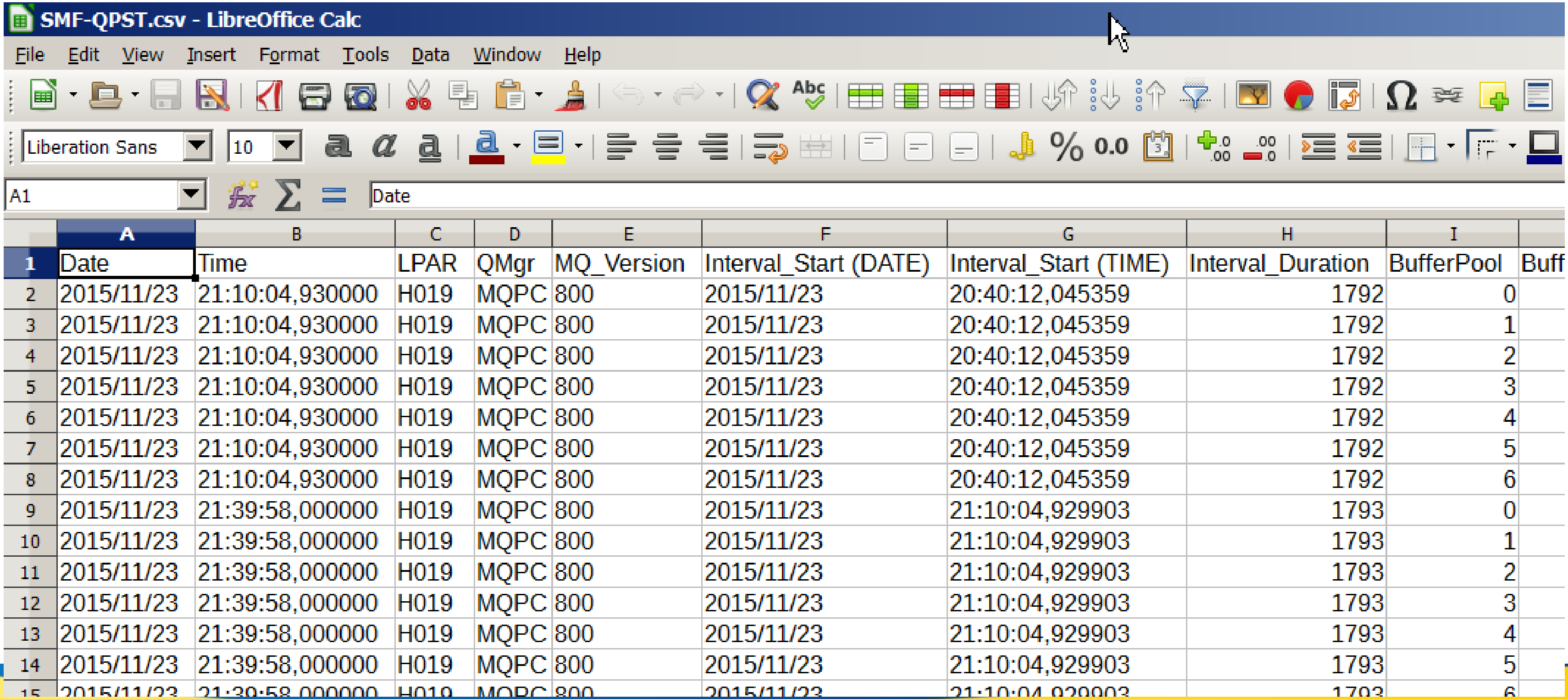
```
C:\smf>mqsmfcsv -i c:\smf\data\test.bin -o c:\smf\out -m 200  -s

MQ SMF CSV - Build Jul 17 2016 11:45:19
Swapping bytes in input records
Processed 146 records total
  Ignored                           record count: 2
  Formatted 115 subtype   1 record count: 48
  Formatted 115 subtype   2 record count: 48
  Formatted 115 subtype 215 record count: 48
```

A raw formatted CSV file



```
Date,Time,LPAR,QMgr,MQ_Version,Interval_Start (DATE),Interval_Start (TIME),Inter
val_Duration,BufferPool,Buffer_Count,Lowest_Stealable,Current_Stealable,Getp_Old
_Requests,Getp_New_Requests,DASD_Read,Set_Write_Pages,Pages_Written,DASD_Write,S
ync_Writes,Defer_Write_THold_Reached,Sync_Write_THold_Reached,Buffer_Steals,Buff
er_Steals_Hash_Changes,Suspend_No_Buffers,LOC,FIX,
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,0,50000,49981,49984,966636,0,0,966532,84,31,17,0,0,0,0,0,"Above","Paged"
,
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,1,100000,42308,99487,1771124,328513,0,1736279,111053,6993,41,0,0,0,0,0,"
Above","Paged",
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,2,50000,49999,49999,221,0,0,13,13,13,13,0,0,0,0,0,"Above","Paged",
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,3,100000,98961,99385,284845,56959,0,280496,4117,276,13,0,0,0,0,0,"Above"
,"Paged",
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,4,50000,49999,49999,741,0,0,13,13,13,13,0,0,0,0,0,"Above","Paged",
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,5,100000,99393,99393,516284,103803,0,508323,828,71,13,0,0,0,0,0,"Above",
"Paged",
"2015/11/23","21:10:04,930000","H019","MQPC",="800","2015/11/23","20:40:12,04535
9",1792,6,100000,99992,99999,2229,167,0,764,36,19,13,0,0,0,0,0,"Above","Paged",
@
"SMF-QPST.csv" 553 lines, 87171 characters
```

Imported to a spreadsheet


SMF-QPST.csv - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
Liberation Sans
10
A1
Date

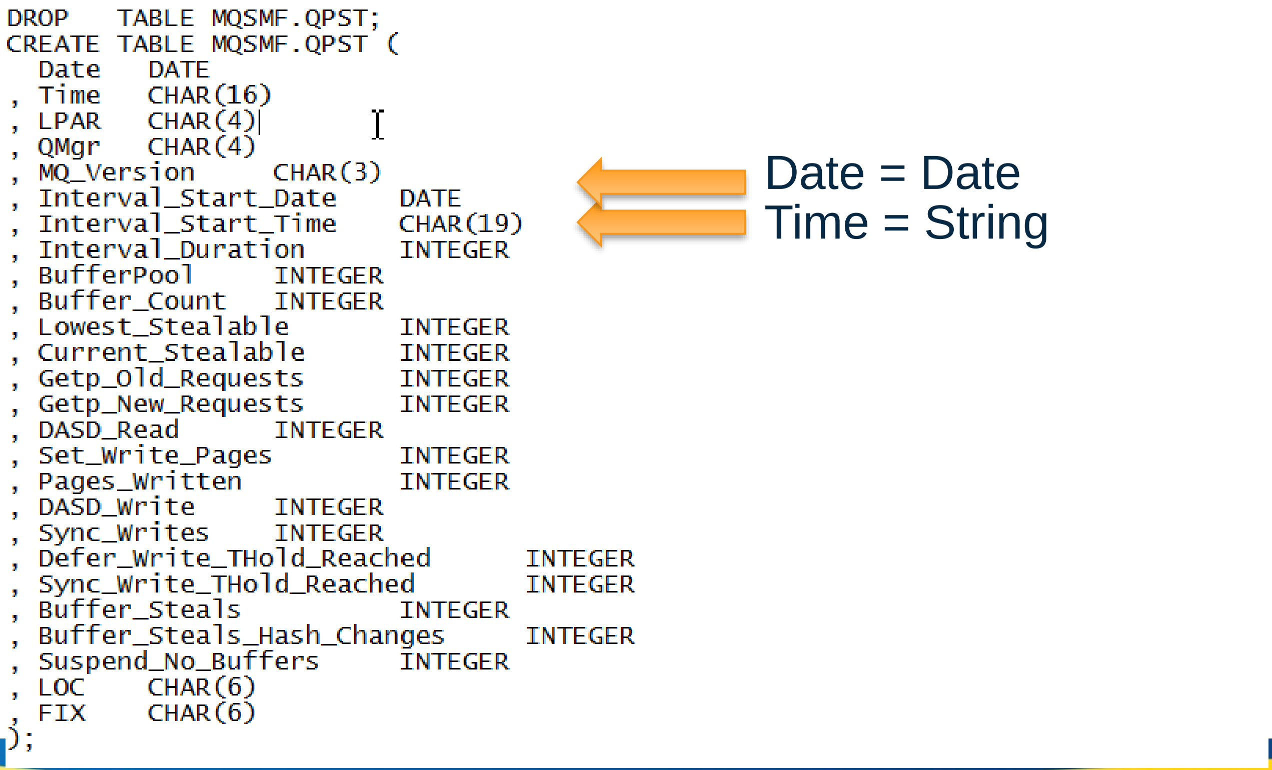
	A	B	C	D	E	F	G	H	I	
1	Date	Time	LPAR	QMgr	MQ_Version	Interval_Start (DATE)	Interval_Start (TIME)	Interval_Duration	BufferPool	Buff
2	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	0	
3	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	1	
4	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	2	
5	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	3	
6	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	4	
7	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	5	
8	2015/11/23	21:10:04,930000	H019	MQPC	800	2015/11/23	20:40:12,045359	1792	6	
9	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	0	
10	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	1	
11	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	2	
12	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	3	
13	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	4	
14	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	5	
15	2015/11/23	21:39:58,000000	H019	MQPC	800	2015/11/23	21:10:04,929903	1793	6	

Import to SQL tables

- After working with just CSV, Lyn tried importing data to DB2
  - For much larger quantities that challenge spreadsheets

- But DB2 cannot simply import CSV files
  - Needs tables to be created with columns and datatypes
  - Unlike MS Access, which does it automatically
- Tried creating tables by hand
  - Was easier to do it in code to cover all tables
  - Get simple DDL to define columns with appropriate types

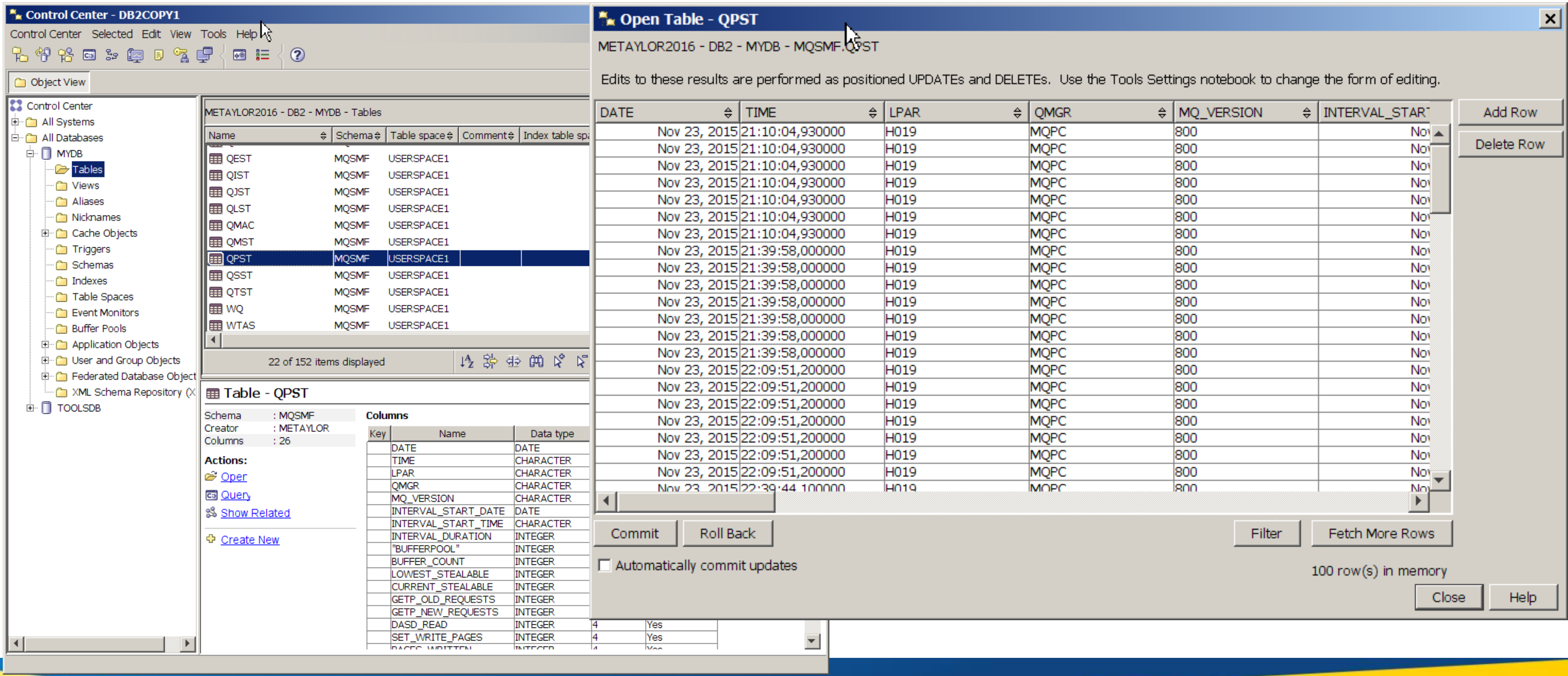
Some DDL



```
DROP   TABLE MQSMF.QPST;
CREATE TABLE MQSMF.QPST (
  Date   DATE
, Time   CHAR(16)
, LPAR   CHAR(4)
, QMgr   CHAR(4)
, MQ_Version     CHAR(3)
, Interval_Start_Date    DATE
, Interval_Start_Time    CHAR(19)
, Interval_Duration      INTEGER
, BufferPool     INTEGER
, Buffer_Count   INTEGER
, Lowest_Stealable       INTEGER
, Current_Stealable      INTEGER
, Getp_Old_Requests      INTEGER
, Getp_New_Requests      INTEGER
, DASD_Read      INTEGER
, Set_Write_Pages        INTEGER
, Pages_Written          INTEGER
, DASD_Write     INTEGER
, Sync_Writes    INTEGER
, Defer_Write_THold_Reached      INTEGER
, Sync_Write_THold_Reached       INTEGER
, Buffer_Steals          INTEGER
, Buffer_Steals_Hash_Changes     INTEGER
, Suspend_No_Buffers     INTEGER
, LOC    CHAR(6)
, FIX    CHAR(6)
);
```

Date = Date
Time = String

How it looks in DB2



Control Center - DB2COPY1

Control Center Selected Edit View Tools Help

Object View

- Control Center
  - All Systems
  - All Databases
    - MYDB
      - Tables
      - Views
      - Aliases
      - Nicknames
      - Cache Objects
      - Triggers
      - Schemas
      - Indexes
      - Table Spaces
      - Event Monitors
      - Buffer Pools
      - Application Objects
      - User and Group Objects
      - Federated Database Object
      - XML Schema Repository (X
    - TOOLSDB

METAYLOR2016 - DB2 - MYDB - Tables

Name	Schema	Table space	Comment	Index table spa
QEST	MQSMF	USERSPACE1		
QIST	MQSMF	USERSPACE1		
QJST	MQSMF	USERSPACE1		
QLST	MQSMF	USERSPACE1		
QMAC	MQSMF	USERSPACE1		
QMST	MQSMF	USERSPACE1		
QPST	MQSMF	USERSPACE1		
QSST	MQSMF	USERSPACE1		
QTST	MQSMF	USERSPACE1		
WQ	MQSMF	USERSPACE1		
WTAS	MQSMF	USERSPACE1		

22 of 152 items displayed

Table - QPST

Schema : MQSMF
Creator : METAYLOR
Columns : 26

Actions:
Oper
Query
Show Related
Create New

Columns

Key	Name	Data type		
	DATE	DATE		
	TIME	CHARACTER		
	LPAR	CHARACTER		
	QMGR	CHARACTER		
	MQ_VERSION	CHARACTER		
	INTERVAL_START_DATE	DATE		
	INTERVAL_START_TIME	CHARACTER		
	INTERVAL_DURATION	INTEGER		
	"BUFFERPOOL"	INTEGER		
	BUFFER_COUNT	INTEGER		
	LOWEST_STEALABLE	INTEGER		
	CURRENT_STEALABLE	INTEGER		
	GETP_OLD_REQUESTS	INTEGER		
	GETP_NEW_REQUESTS	INTEGER		
	DASD_READ	INTEGER	4	Yes
	SET_WRITE_PAGES	INTEGER	4	Yes

Open Table - QPST

METAYLOR2016 - DB2 - MYDB - MQSMF.QPST

Edits to these results are performed as positioned UPDATEs and DELETEs. Use the Tools Settings notebook to change the form of editing.

DATE	TIME	LPAR	QMGR	MQ_VERSION	INTERVAL_STAR
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:10:04,930000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	21:39:58,000000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No
Nov 23, 2015	22:09:51,200000	H019	MQPC	800	No

Add Row Delete Row

Commit Roll Back Filter Fetch More Rows

☐ Automatically commit updates

100 row(s) in memory

Close Help

And now some examples of using the data

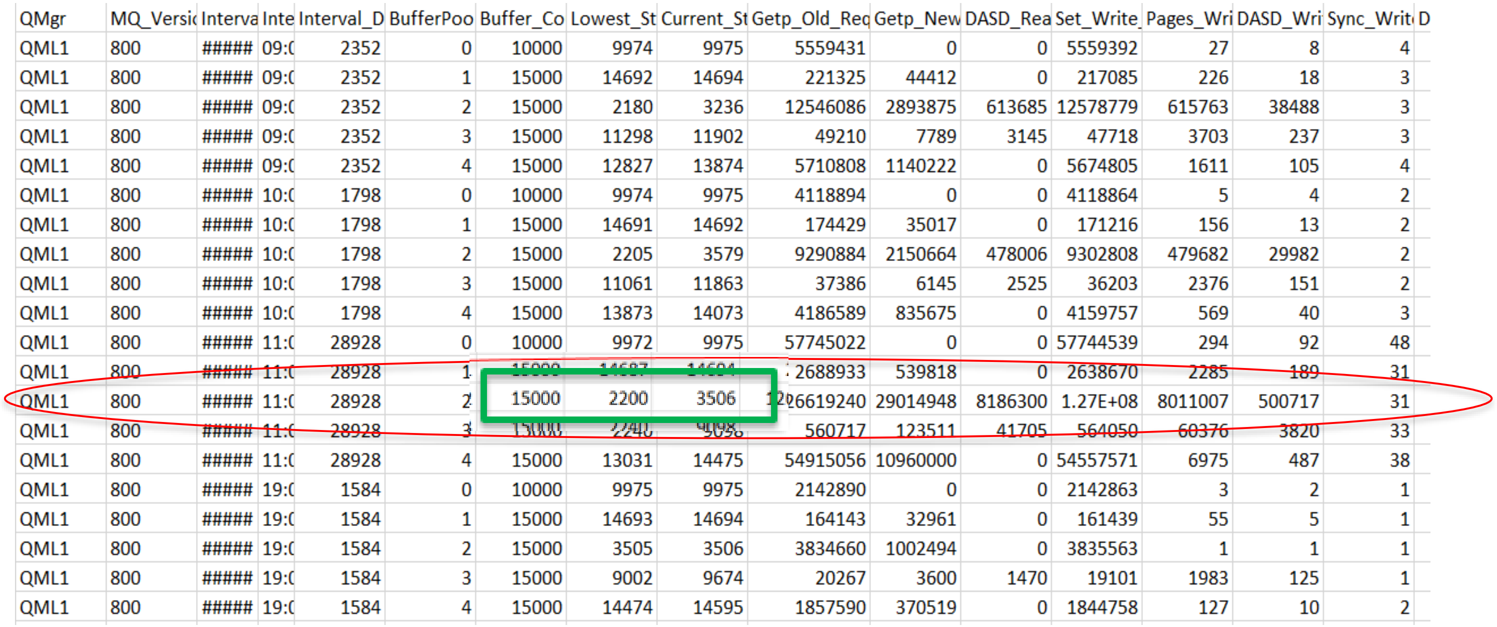
How do I use this?

- Use MP1B and mqsmfcsv together for fuller picture

- MP1B:
  - Looking at messages
  - Examine complete task record
    - What queues used
- MQSMFCSV
  - Looking for specifics

Some Common Analysis

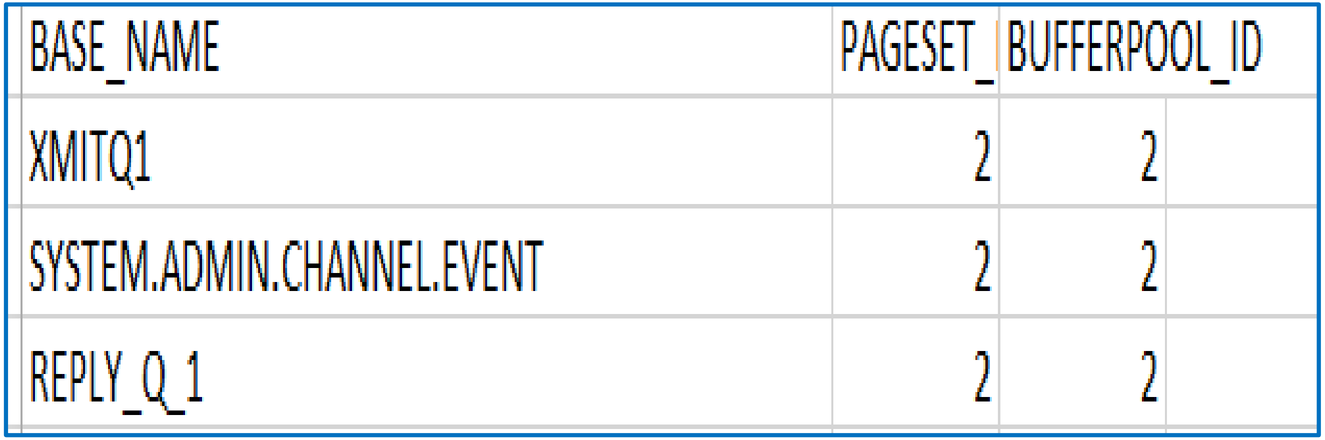
- Bufferpool issues:



QMgr	MQ_Versi	Interva	Inte	Interval_D	BufferPoo	Buffer_Co	Lowest_St	Current_St	Getp_Old_Req	Getp_New	DASD_Rea	Set_Write_	Pages_Wri	DASD_Wri	Sync_Writ	D
QML1	800	#####	09:(	2352	0	10000	9974	9975	5559431	0	0	5559392	27	8	4	
QML1	800	#####	09:(	2352	1	15000	14692	14694	221325	44412	0	217085	226	18	3	
QML1	800	#####	09:(	2352	2	15000	2180	3236	12546086	2893875	613685	12578779	615763	38488	3	
QML1	800	#####	09:(	2352	3	15000	11298	11902	49210	7789	3145	47718	3703	237	3	
QML1	800	#####	09:(	2352	4	15000	12827	13874	5710808	1140222	0	5674805	1611	105	4	
QML1	800	#####	10:(	1798	0	10000	9974	9975	4118894	0	0	4118864	5	4	2	
QML1	800	#####	10:(	1798	1	15000	14691	14692	174429	35017	0	171216	156	13	2	
QML1	800	#####	10:(	1798	2	15000	2205	3579	9290884	2150664	478006	9302808	479682	29982	2	
QML1	800	#####	10:(	1798	3	15000	11061	11863	37386	6145	2525	36203	2376	151	2	
QML1	800	#####	10:(	1798	4	15000	13873	14073	4186589	835675	0	4159757	569	40	3	
QML1	800	#####	11:(	28928	0	10000	9972	9975	57745022	0	0	57744539	294	92	48	
QML1	800	#####	11:(	28928	1	15000	14687	14694	2688933	539818	0	2638670	2285	189	31	
QML1	800	#####	11:(	28928	2	15000	2200	3506	126619240	29014948	8186300	1.27E+08	8011007	500717	31	
QML1	800	#####	11:(	28928	3	15000	2240	9098	560717	123511	41705	564050	60376	3820	33	
QML1	800	#####	11:(	28928	4	15000	13031	14475	54915056	10960000	0	54557571	6975	487	38	
QML1	800	#####	19:(	1584	0	10000	9975	9975	2142890	0	0	2142863	3	2	1	
QML1	800	#####	19:(	1584	1	15000	14693	14694	164143	32961	0	161439	55	5	1	
QML1	800	#####	19:(	1584	2	15000	3505	3506	3834660	1002494	0	3835563	1	1	1	
QML1	800	#####	19:(	1584	3	15000	9002	9674	20267	3600	1470	19101	1983	125	1	
QML1	800	#####	19:(	1584	4	15000	14474	14595	1857590	370519	0	1844758	127	10	2	

Some Common Analysis

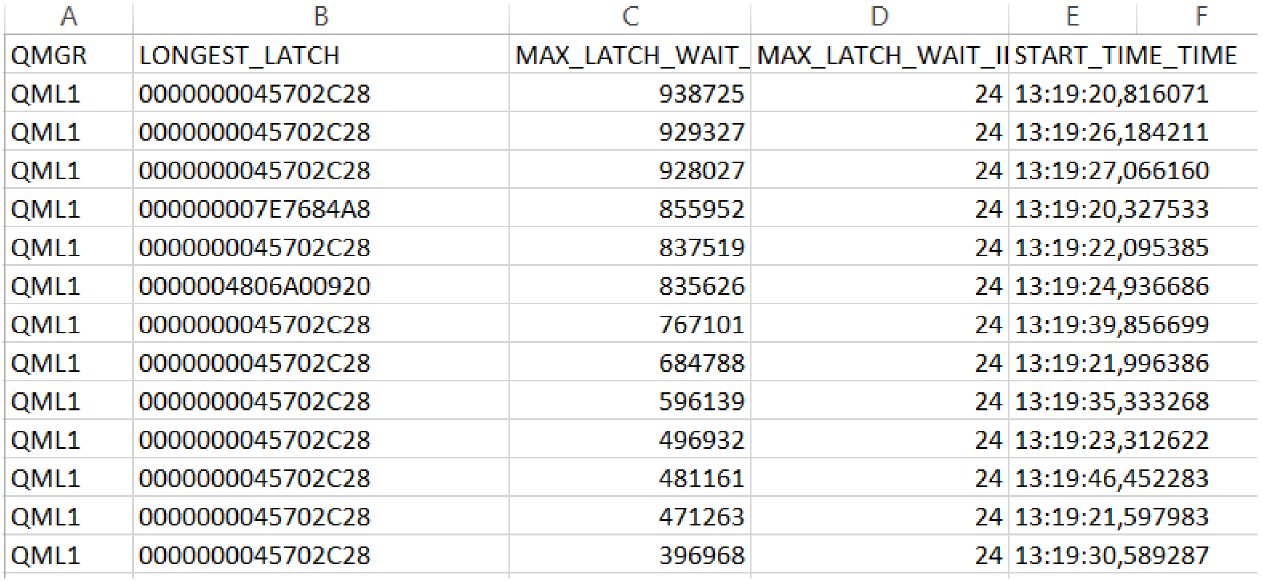
- Who is using the bufferpool?



BASE_NAME	PAGESET_	BUFFERPOOL_ID
XMITQ1	2	2
SYSTEM.ADMIN.CHANNEL.EVENT	2	2
REPLY_Q_1	2	2

Some Common Analysis

- Long Latching:



A	B	C	D	E	F
QMGR	LONGEST_LATCH	MAX_LATCH_WAIT_	MAX_LATCH_WAIT_I	START_TIME_TIME	
QML1	000000045702C28	938725	24	13:19:20,816071	
QML1	000000045702C28	929327	24	13:19:26,184211	
QML1	000000045702C28	928027	24	13:19:27,066160	
QML1	000000007E7684A8	855952	24	13:19:20,327533	
QML1	000000045702C28	837519	24	13:19:22,095385	
QML1	0000004806A00920	835626	24	13:19:24,936686	
QML1	000000045702C28	767101	24	13:19:39,856699	
QML1	000000045702C28	684788	24	13:19:21,996386	
QML1	000000045702C28	596139	24	13:19:35,333268	
QML1	000000045702C28	496932	24	13:19:23,312622	
QML1	000000045702C28	481161	24	13:19:46,452283	
QML1	000000045702C28	471263	24	13:19:21,597983	
QML1	000000045702C28	396968	24	13:19:30,589287	

Queries against the data

- Reading a million-plus row report for potential issues is impossible
  - With V7.0.1 we developed a series of searches that worked well against the task report
  - Quit working with V7.1 because the format changed dramatically
- Using queries to find things which might be problems

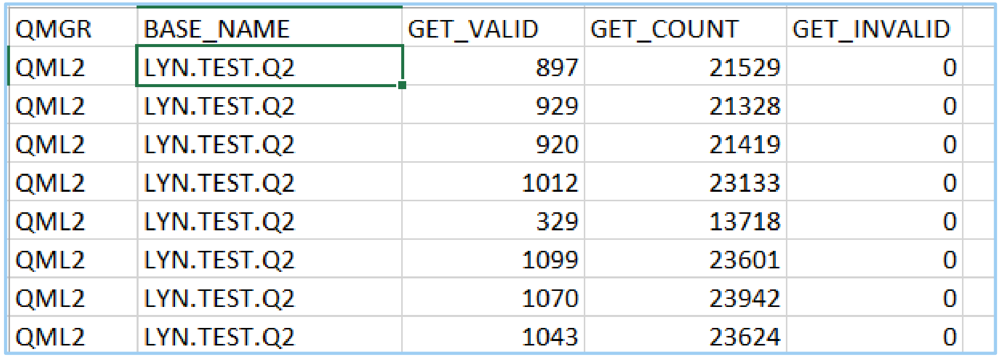
Some queries I have found useful (to date)

- Looking for skipped or expired messages?
  - SELECT LPAR, QMgr, Correlation,Base_Name from MQSMF.WQ WHERE Get_Messages_Skipped_Count >0;
  - SELECT LPAR, QMgr, Correlation,Base_Name from MQSMF.WQ WHERE Get_Messages_Expired_Count >0;

- Put to waiting getter active on a queue?
  - SELECT * from MQSMF.WQ   WHERE LPAR = ‘MPX1' AND "Base_Name" = ‘LYNS.TEST.QUEUE' AND "Put_Waiting_Getter_Count" > 0 ;

Some queries I have found useful (to date)

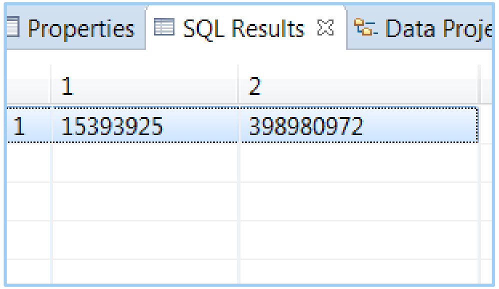
- How many transactions had unfulfilled MQGETs?
  - SELECT QMGR, Base_Name, Get_Valid, Get_Count, Get_Invalid from MQSMF.WQ where (GET_Valid < Get_Count and Base_Name= ‘LYN.TEST.Q2');



QMGR	BASE_NAME	GET_VALID	GET_COUNT	GET_INVALID
QML2	LYN.TEST.Q2	897	21529	0
QML2	LYN.TEST.Q2	929	21328	0
QML2	LYN.TEST.Q2	920	21419	0
QML2	LYN.TEST.Q2	1012	23133	0
QML2	LYN.TEST.Q2	329	13718	0
QML2	LYN.TEST.Q2	1099	23601	0
QML2	LYN.TEST.Q2	1070	23942	0
QML2	LYN.TEST.Q2	1043	23624	0

Some queries I have found useful (to date)

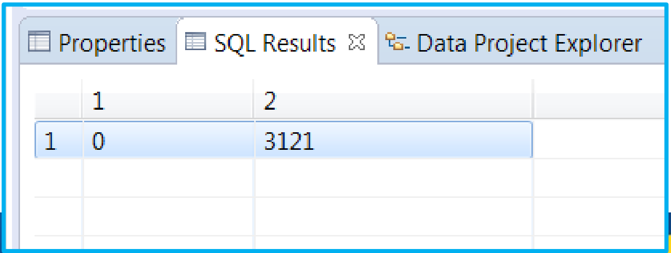
- How many valid MQGETs were from a queue?
  - SELECT SUM(Get_Valid), SUM(Get_Count) from MQSMF.WQ where (GET_Valid < Get_Count and Base_Name= 'LYNE.QUEUE.2');
  - Results - Column 1 the number of valid gets, Column 2 is total get requests:



	1	2
1	15393925	398980972

Some queries I have found useful (to date)

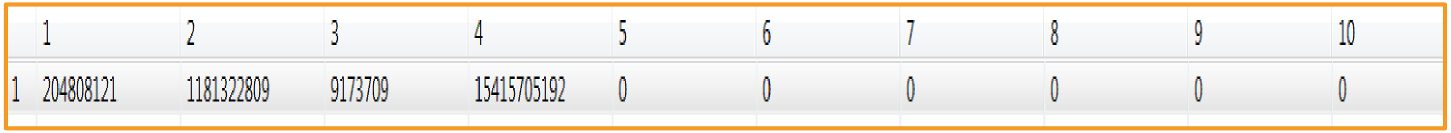
- What was my largest message size retrieved for this queue?
  - SELECT MAX(Get_Max_Msg_Size) from MQSMF.WQ where (Base_Name= ‘LYNS.TEST.QUEUE');
  - Result was 11,189 (application people insisted it was 3,800)
- How many MQPUTs and MQPUT1s were completed?
  - SELECT SUM (Put_Count), SUM (Put1_Count) from MQSMF.WQ where (Base_Name = ‘LYNS.TEST.QUEUE');
  - Results:



	1	2
1	0	3121

Useful Queries - How much are my puts and gets costing?

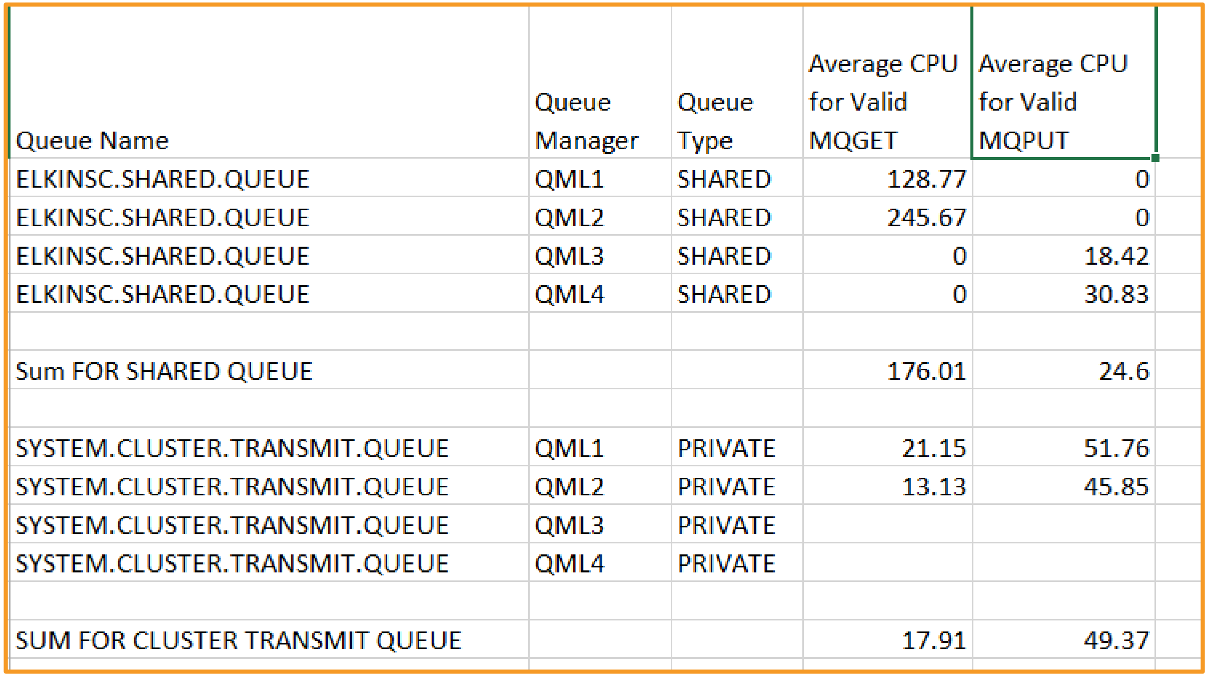
- Query to get total costs for MQGETs and MQPUTs
  -
```
SELECT SUM (Get_Count),  SUM (Get_CT_us), SUM (Total_Valid_Gets),
                 SUM (Total_Bytes_Get),
                 SUM (Put_Count), SUM (Put_CT_us), SUM (Put1_Count),
                 SUM (Put1_CT_us), SUM (Total_Valid_Puts), SUM (Total_Bytes_Put)
   FROM MQSMF.WQ
   WHERE (Base_Name = ‘ELKINSC.SHARED.QUEUE' AND
                 QMGR = 'QML1');
```



	1	2	3	4	5	6	7	8	9	10
1	204808121	1181322809	9173709	15415705192	0	0	0	0	0	0

Useful Queries - How much are my puts and gets costing?

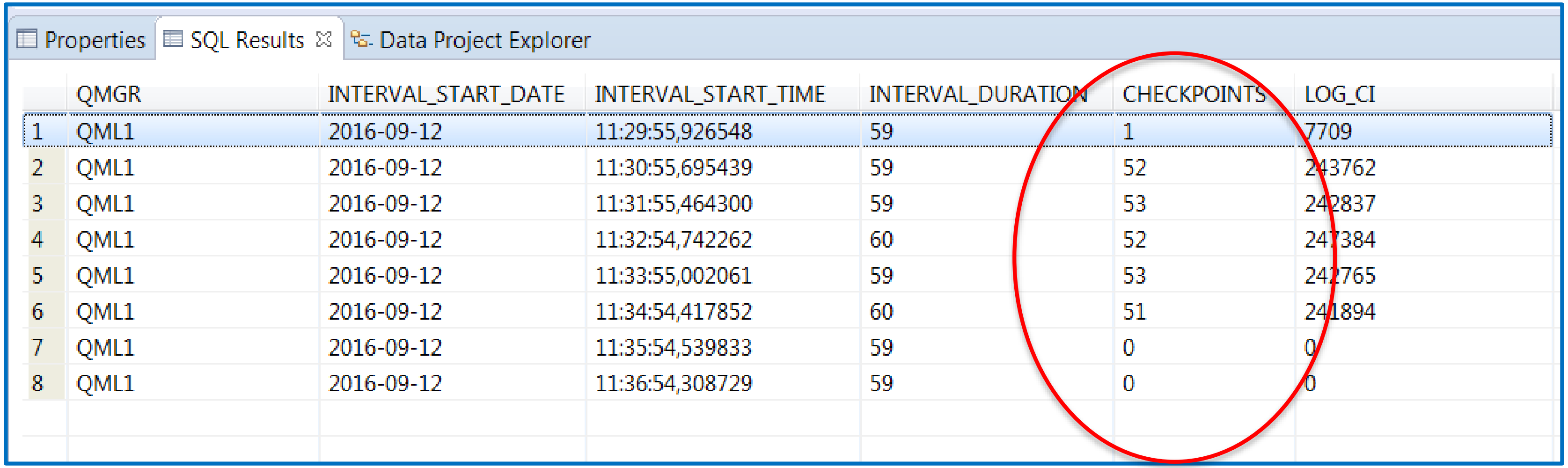
- The raw sums are not all that useful by themselves
- But when averaged and used for comparisons can be



Queue Name	Queue Manager	Queue Type	Average CPU for Valid MQGET	Average CPU for Valid MQPUT
ELKINSC.SHARED.QUEUE	QML1	SHARED	128.77	0
ELKINSC.SHARED.QUEUE	QML2	SHARED	245.67	0
ELKINSC.SHARED.QUEUE	QML3	SHARED	0	18.42
ELKINSC.SHARED.QUEUE	QML4	SHARED	0	30.83
Sum FOR SHARED QUEUE			176.01	24.6
SYSTEM.CLUSTER.TRANSMIT.QUEUE	QML1	PRIVATE	21.15	51.76
SYSTEM.CLUSTER.TRANSMIT.QUEUE	QML2	PRIVATE	13.13	45.85
SYSTEM.CLUSTER.TRANSMIT.QUEUE	QML3	PRIVATE		
SYSTEM.CLUSTER.TRANSMIT.QUEUE	QML4	PRIVATE		
SUM FOR CLUSTER TRANSMIT QUEUE			17.91	49.37

And can surprise you!

- SELECT QMgr, Interval_Start_Date, Interval_Start_Time,
  Interval_Duration, Checkpoints, Log_CI
  FROM MQSMF.QJST;

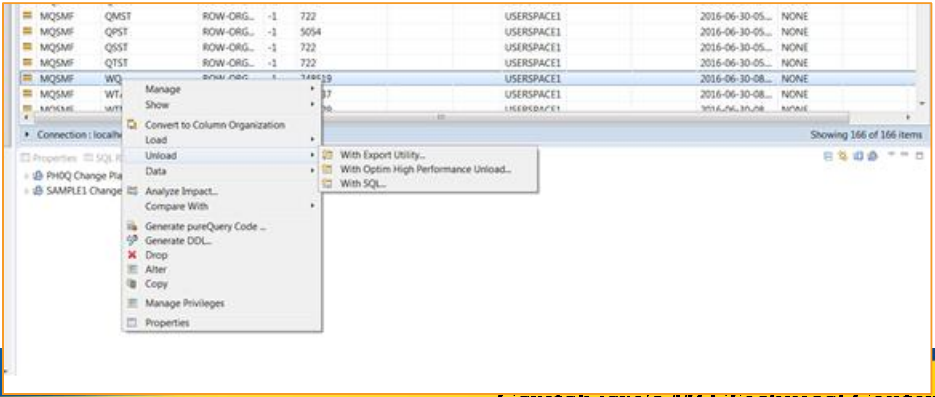


Properties | SQL Results | Data Project Explorer

	QMGR	INTERVAL_START_DATE	INTERVAL_START_TIME	INTERVAL_DURATION	CHECKPOINTS	LOG_CI
1	QML1	2016-09-12	11:29:55,926548	59	1	7709
2	QML1	2016-09-12	11:30:55,695439	59	52	243762
3	QML1	2016-09-12	11:31:55,464300	59	53	242837
4	QML1	2016-09-12	11:32:54,742262	60	52	247384
5	QML1	2016-09-12	11:33:55,002061	59	53	242765
6	QML1	2016-09-12	11:34:54,417852	60	51	241894
7	QML1	2016-09-12	11:35:54,539833	59	0	0
8	QML1	2016-09-12	11:36:54,308729	59	0	0

Queries – warnings and lessons learned

- Using Data Studio
  - Makes things easy for those of us who are not very SQL literate
  - Single quotes are typically used for literals
  - A query defaults to 500 rows
    - If you need to see more, use an EXPORT
- Using an EXPORT

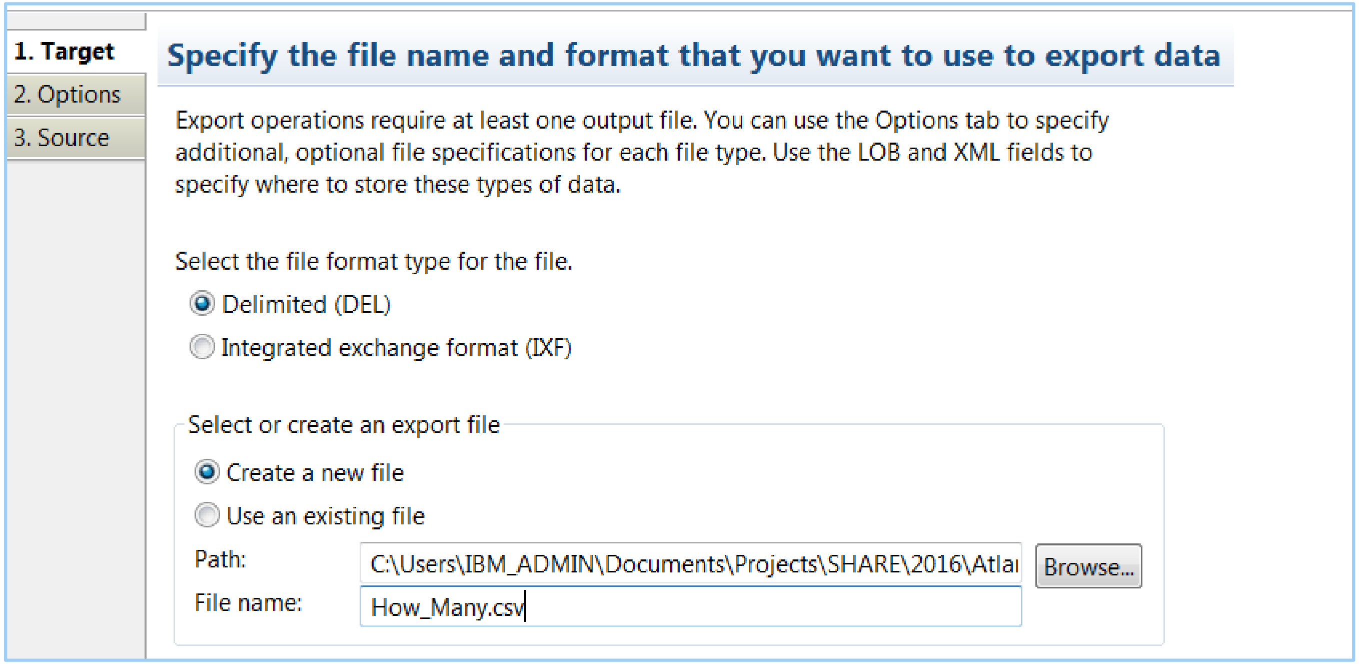


MQSMF	QMST	ROW-ORG...	-1	722	USERSPACE1	2016-06-30-05...	NONE
MQSMF	QPST	ROW-ORG...	-1	5054	USERSPACE1	2016-06-30-05...	NONE
MQSMF	QSST	ROW-ORG...	-1	722	USERSPACE1	2016-06-30-05...	NONE
MQSMF	QTST	ROW-ORG...	-1	722	USERSPACE1	2016-06-30-05...	NONE
MQSMF	WQ				USERSPACE1	2016-06-30-08...	NONE
MQSMF	WT				USERSPACE1	2016-06-30-08...	NONE

Connection : localh

Showing 166 of 166 items

Using an Export - continued



1. Target
2. Options
3. Source

Specify the file name and format that you want to use to export data

Export operations require at least one output file. You can use the Options tab to specify additional, optional file specifications for each file type. Use the LOB and XML fields to specify where to store these types of data.

Select the file format type for the file.

- [✓ Delimited (DEL)]
- Integrated exchange format (IXF)

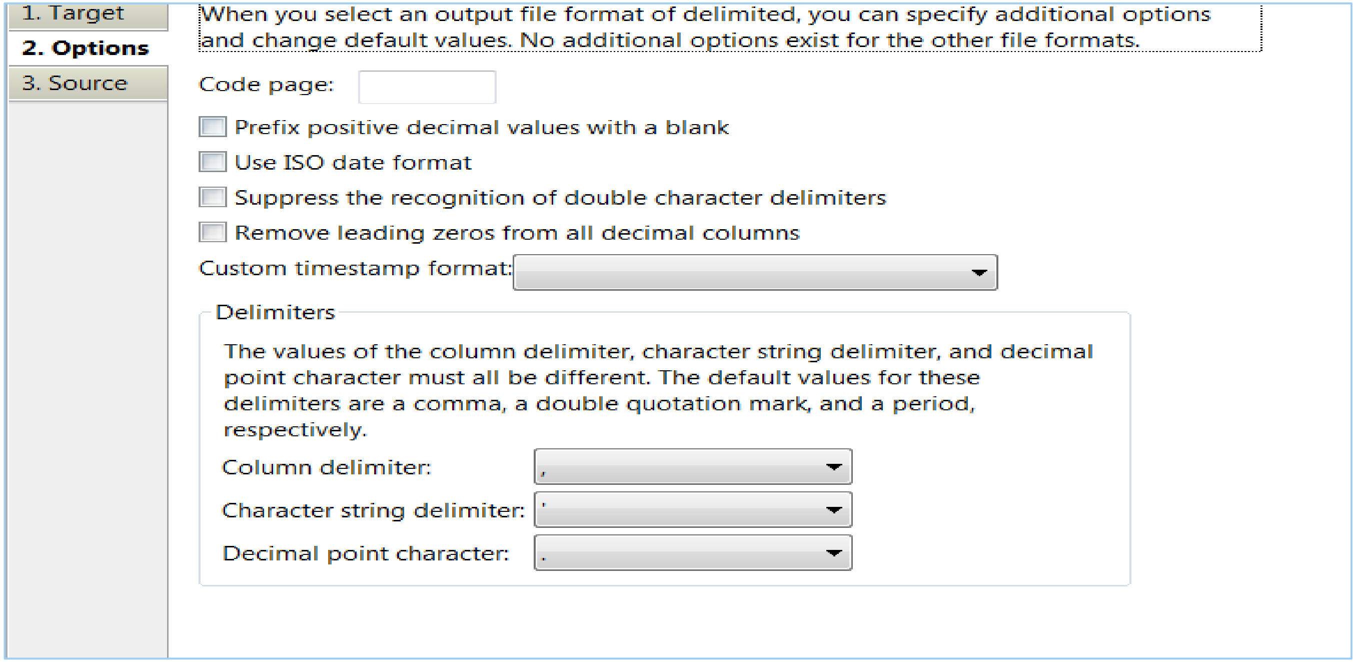
Select or create an export file

- [✓ Create a new file]
- Use an existing file

Path: C:\Users\IBM_ADMIN\Documents\Projects\SHARE\2016\Atla

File name: How_Many.csv

Using an Export - continued



1. Target
2. Options
3. Source

When you select an output file format of delimited, you can specify additional options and change default values. No additional options exist for the other file formats.

Code page:

- [ ] Prefix positive decimal values with a blank
- [ ] Use ISO date format
- [ ] Suppress the recognition of double character delimiters
- [ ] Remove leading zeros from all decimal columns

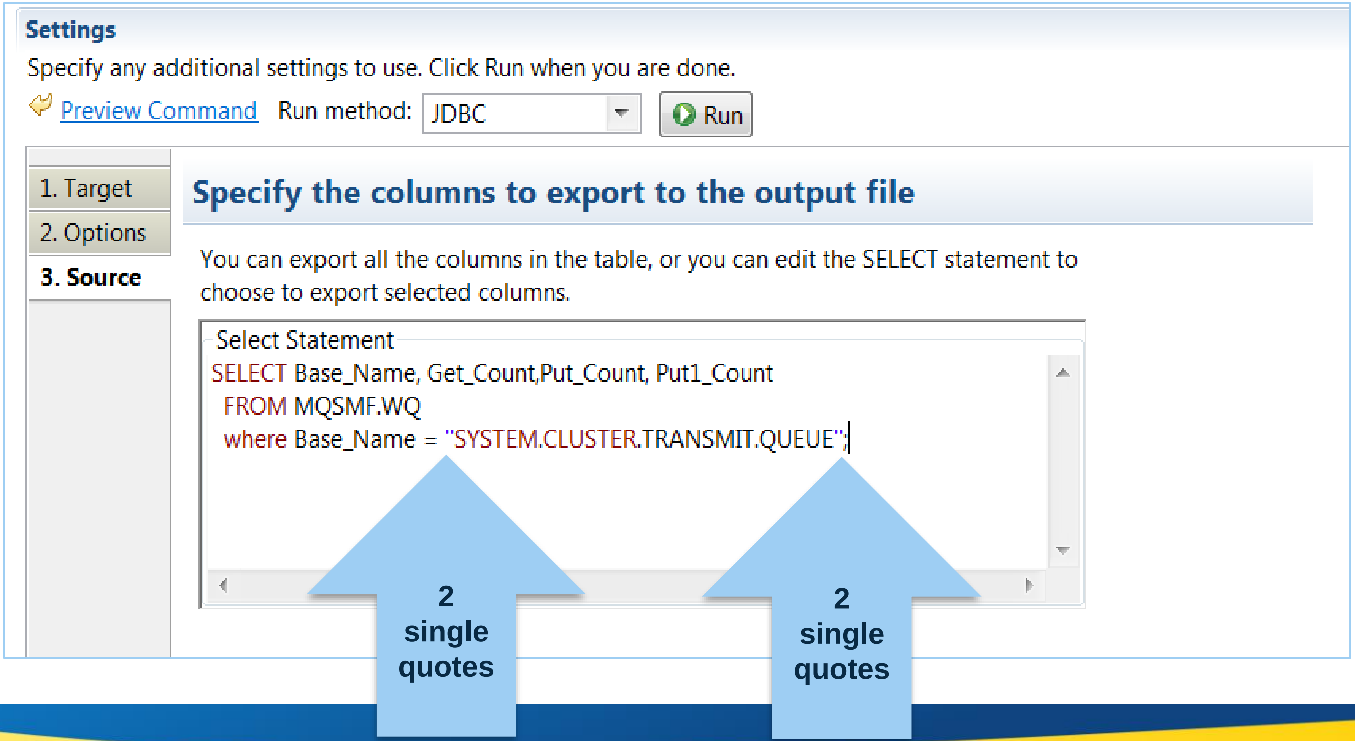
Custom timestamp format:

Delimiters

The values of the column delimiter, character string delimiter, and decimal point character must all be different. The default values for these delimiters are a comma, a double quotation mark, and a period, respectively.

Column delimiter:	,
Character string delimiter:	"
Decimal point character:	.

Using an Export - continued



Settings

Specify any additional settings to use. Click Run when you are done.

Preview Command Run method: JDBC Run

Specify the columns to export to the output file

You can export all the columns in the table, or you can edit the SELECT statement to choose to export selected columns.

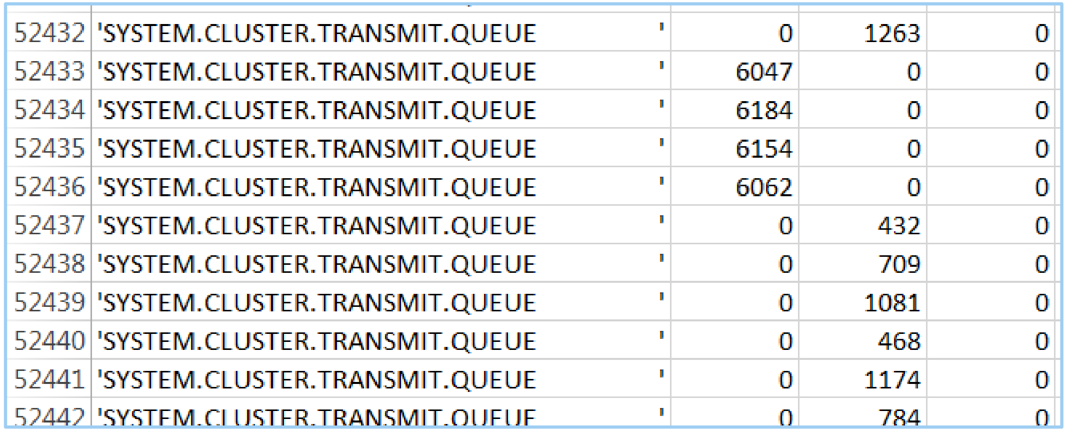
Select Statement

```
SELECT Base_Name, Get_Count,Put_Count, Put1_Count
 FROM MQSMF.WQ
 where Base_Name = "SYSTEM.CLUSTER.TRANSMIT.QUEUE";
```

The Export Command generated

```
CALL SYSPROC.ADMIN_CMD(
'EXPORT TO "C:\Users\IBM_ADMIN\Documents\Projects\SHARE\2016\Atlanta\How_Many.csv"
OF DEL MODIFIED BY COLDEL, CHARDEL'''' DECPT.
MESSAGES ON SERVER
SELECT Base_Name, Get_Count,Put_Count, Put1_Count
  FROM MQSMF.WQ
  where Base_Name = ''SYSTEM.CLUSTER.TRANSMIT.QUEUE'';' );
```

Results of the export (end of CSV file)



52432	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	1263	0
52433	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	6047	0	0
52434	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	6184	0	0
52435	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	6154	0	0
52436	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	6062	0	0
52437	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	432	0
52438	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	709	0
52439	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	1081	0
52440	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	468	0
52441	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	1174	0
52442	'SYSTEM.CLUSTER.TRANSMIT.QUEUE '	0	784	0

Summary

- MQ’s SMF provides much insight for tuning and planning
- Experience has been needed to analyse data
- The discussion of tooling and queries here should enable better self-service

Any questions?