# *New MQ CHINIT Monitoring via SMF (z/OS)*

**Lyn Elkins – elkinsc@us.ibm.com**

# Agenda

- **CHINIT SMF**
  - Channel Initiator Statistics
  - Channel Accounting Data

# CHINIT SMF: The Problem

- Prior to MQ v8.0, there was limited SMF data for channels
- With CLASS(3) ACCOUNTING trace:

**START TRACE(ACCTG) DEST(SMF) CLASS(3)**

**You get the Task/Thread Identification (WTID) SMF 116 Subtype 1 record which gives you data about the Sender or Receiver Message Channel Agent thread:**

```
      ====> New task record found     <==========
Thread type.............> MSVER
Connection name.........> QML4CHIN
Operator ID.............> MQUSER
User ID.................> MQUSER
Channel name............> QSGM.OUT
Chl connection..........> 1.2.3.43
Correlator ID...........>             }
Correlator ID.....(HEX)> 243DD000E7E75C5C243DD2C0
Context token...........>
Context token.....(HEX)> 00000000000000000000000000000000
NID                    > QML4CHING  W È¢
```

CHINIT TASK

Channel Name and Connection

# CHINIT SMF: The Problem

- **So, prior to MQ v8.0, there was no detailed, useful data for:**
  - CHINIT address space
  - Channel activity
- **Many customers have had to create their own ‘monitoring’ jobs**
  - They issues periodic **DISPLAY CHSTATUS** commands
  - Or use the **MQCMD** program from SupportPac MP1B to do this
- **Difficult to:**
  - Monitor activity in the CHINIT address space
    - Investigate performance issues and tune for better performance
    - Perform capacity planning
    - Manage historical data

# CHINIT SMF: The Problem

- **The category 2 SupportPac MP1B provides a program called MQCMD which can be used to automate issuing DISPLAY commands on a regular basis, which facilitates monitoring of channels. The output is formatted using Comma Separated Values (CSVs) for ease of importing into a spreadsheet for analysis.**

# CHINIT SMF: The Solution

- **Channel Initiator Statistics**
  - High level view of activity in the CHINIT address space
    - Data about Dispatcher tasks
      - » Number of channels running, TCB usage
    - Data about Adapter, DNS and SSL tasks
  - Used to:
    - Determine if there is spare capacity
    - More effective tuning of dispatcher and adapter tasks
- **Channel Accounting Data**
  - Detailed view of individual channels
    - What work are the channels doing ?
    - Which channels are heavily utilized ?

# Channel Initiator Statistics

- **Channel initiator**
  - QSG name
  - Number of current channels
  - Maximum current channels
  - Number of active channels
  - Maximum active channels
  - Maximum TCP/IP channels
  - Maximum LU 6.2 channels
  - Storage usage in MB

- Dispatcher task
  - Task number (TCB address)
  - Number of requests for task
  - Busy CPU time of task
  - Sum of elapsed time of requests
  - Wait elapsed time of task

- Adapter task
  - Task number (TCB address)
  - Number of requests for task
  - Busy CPU time of task
  - Sum of elapsed time of requests
  - Wait elapsed time of task

**DNS task**

Task number (TCB address)
Number of requests for task
Busy CPU time of task
Sum of elapsed time of requests
Wait elapsed time of task
Time of day of max DNS request
Duration time of max DNS request

**SSL task**

Task number (TCB address)
Number of requests for task
Busy CPU time of task
Sum of elapsed time of requests
Wait elapsed time of task
Time of day of max SSL request
Duration of max SSL request

# Channel Accounting Data

- **For each channel instance**
  - Channel name
  - Channel disposition
  - Channel type
  - Channel state
  - STATCHL setting
  - Connection name
  - Channel stopped date & time
  - Last msg date & time
  - Channel batch size
  - Num of messages
  - Num of persistent messages
  - Num of batches
  - Num of full batches
  - Num of transmission buffers sent
  - Num of transmission buffers received
  - Current shared conversations
  - Num of bytes
  - Number of persistent bytes
  - Number of bytes sent (both ctrl data & msg data)
  - Number of bytes received (both ctrl data & msg data)
  - Compression rate
  - Exit time average
  - Exit time min
  - Exit time max
  - Exit time max date & time
  - Net time average
  - Net time min
  - Net time max
  - Net time max date & time
  - Remote qmgr /app name
  - Put retry count
  - Transmission queue empty count

# New SMF record subtypes and DSECTs

- **New subtypes**

  **SMF 115 subtype 231** (0xE7=‘X’) for Channel Initiator Statistics

  **SMF 116 subtype 10** for Channel Accounting Data

- **New DSECTs shipped**

  **CSQDQWHS (QWHS)**: Standard header

  **CSQDQWSX (QWSX)**: Self defining section for **subtype 231**
  **CSQDQCCT (QCCT)**: Definition for CHINIT statistics data
  - **CSQDQCT (QCT_DSP/QCT_ADP/QCT_SSL/QCT_DNS)**: Definition for CHINIT tasks

  **CSQDQHS (QWHS)**: Standard header
  **CSQDQWS5 (QWS5)**: Self defining section for **subtype 10**

  **CSQDQCST (QCST)**: Definition for channel accounting data

# New SMF record subtypes and DSECTs - Notes

Two new SMF records have been added:

SMF 115 sub type 231 has the CHINIT control information like adapter and dispatcher task CPU times, DNS resolution times. This helps with tuning the number of tasks configured.

SMF 116 sub type 10 has the per channel accounting data like bytes sent, achieved batch size, etc.

The DSECTs that are shipped for each type of record are listed.

Note: The standard layout for SMF records involves three parts:

- SMF header - Provides format, identification, and time and date information about the record itself.
- Self-defining section - Defines the location and size of the individual data records within the SMF record.
- Data records - The actual data from MQ that you want to analyze.

# Starting CHINIT SMF

- **Before starting the statistics trace, the DISPLAY TRACE output may look something like this:**

```
RESPONSE=MPX1
 CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
 TNO TYPE   CLASS        DEST     USERID   RMID
 01  GLOBAL 01           RES      *        *
 02  STAT   01,02        SMF      *        *
 04  ACCTG  03           SMF      *        *
 00  CHINIT *            RES      *        *
 END OF TRACE REPORT
 CSQ9022I QML1 CSQWVCM1 ' DISPLAY TRACE' NORMAL COMPLETION
```

- **Start the Channel Initiator Statistics via the ‘START TRACE’ command:**
  - +cpf START TRACE(STAT) CLASS(4)

# Starting CHINIT SMF - continued

- The START TRACE response should look as shown:

```
CSQW130I QML1 'STAT' TRACE STARTED, ASSIGNED TRACE NUMBER 03
CSQ9022I QML1 CSQWVCM1 ' START TRACE' NORMAL COMPLETION
```

- The DISPLAY TRACE output should look something like this:

```
RESPONSE=MPX1
 CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
 TNO TYPE   CLASS        DEST     USERID   RMID
 01  GLOBAL 01           RES      *        *
 02  STAT   01,02        SMF      *        *
 03  STAT   04           SMF      *        *
 04  ACCTG  03           SMF      *        *
 00  CHINIT *            RES      *        *
 END OF TRACE REPORT
```

# Starting Channel Accounting SMF

- **Start the Channel Accounting SMF via the ‘START TRACE’ command:**
  - +cpf START TRACE(ACCTG) CLASS(4)
  - The result from the start command should look something like this:

```
CSQW130I QML1 'ACCTG' TRACE STARTED, ASSIGNED TRACE NUMBER 05

CSQ9022I QML1 CSQWVCM1 ' START TRACE' NORMAL COMPLETION
```

# Starting Channel Accounting SMF - continued

- **The DISPLAY TRACE output should look something like this:**

```
RESPONSE=MPX1
 CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
 TNO TYPE   CLASS        DEST     USERID   RMID
 01  GLOBAL 01           RES      *        *
 02  STAT   01,02        SMF      *        *
 03  STAT   04           SMF      *        *
 04  ACCTG  03           SMF      *        *
 05  ACCTG  04           SMF      *        *
 00  CHINIT *            RES      *        *
 END OF TRACE REPORT
```

# Starting CHINIT SMF automatically

- **The CSQ6SYSP macro parameters SMFSTAT and SMFACCT have been extended:**
  - **SMFSTAT – now accepts a ‘4’ to automatically start the CHINIT statistics**
  - **SMFACCT – now accepts a ‘4’ to automatically start the channel accounting**
  - **SMF is started when the channel initiator is started**

- **“Can be disabled/re-enabled by STOP/START TRACE while CHINIT started”**
  - This is not currently true

# Starting CHINIT SMF automatically - continued

- Setting SMFSTAT and SMFACCT to ‘4’ results in the following:

```
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -  089
TNO TYPE   CLASS        DEST     USERID   RMID
01  GLOBAL 01           RES      *        *
02  STAT   04           SMF      *        *
03  ACCTG  04           SMF      *        *
00  CHINIT *            RES      *        *
END OF TRACE REPORT
CSQ9022I QML1 CSQWVCM1 ' DISPLAY TRACE' NORMAL COMPLETION
```

- The SMF data only includes the new SMF 115 and 116 Subtypes:

```
Summary of MQ SMF records and subtypes found
============================================
SMF type 115 subtype 231, record count       3 Chinit
SMF type 116 subtype  10, record count       3 Channel data
```

- Which is probably not what was intended.

# Starting CHINIT SMF automatically – getting more than just the CHINIT data

- **The CSQ6SYSP macro parameters SMFSTAT and SMFACCT have been extended:**
  - **SMFSTAT & SMFACCT – the traces can be 'stacked' in the macro as shown:**

```
SMFACCT=(01,03,04),   GATHER SMF ACCOUNTING          X
SMFSTAT=(01,04),      GATHER SMF STATS               X
```

  - **Note that using the (01:04) value is not allowed in the SYSP macro. It is on the START TRACE command.**

# Starting CHINIT SMF automatically - continued

- **Setting SMFSTAT to (01,04) and SMFACCT to (01,03,04) results in the following:**

```
RESPONSE=MPX1
 CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
 TNO TYPE    CLASS         DEST     USERID   RMID
 01  GLOBAL  01            RES      *        *
 02  STAT    01,04         SMF      *        *
 03  ACCTG   01,03,04      SMF      *        *
 00  CHINIT  *             RES      *        *
 END OF TRACE REPORT
```

- **The SMF data now includes all the SMF 115 and 116 data:**

```
Summary of MQ SMF records and subtypes found
============================================
SMF type 115 subtype   1, record count      14 System statistics(1)
SMF type 115 subtype   2, record count      14 System statistics(2)
SMF type 115 subtype 215, record count      14 Buffer manager extension
SMF type 115 subtype 231, record count      27 Chinit
SMF type 116 subtype   0, record count      13 Accounting class(1)
SMF type 116 subtype   1, record count     140 Accounting class(3)
SMF type 116 subtype  10, record count      27 Channel data
```

# CHINIT SMF: Controls

- STAT trace allows a high level view of activity in the CHINIT address space.
- ACCTG trace allows a detailed view at the channel level.

# New console messages for CHINIT SMF

- **For START/STOP TRACE(STAT)**

  **CSQX128I csect-name Channel initiator statistics collection started**

  **CSQX129I csect-name Channel initiator statistics collection stopped**

- **For START/STOP TRACE(ACCTG)**

  **CSQX126I csect-name Channel accounting collection started**

  **CSQX127I csect-name Channel accounting collection stopped**

# Controlling the CHINIT SMF interval

- **The STATIME parameter controls the interval for everything**
  - Controls the SMF interval for both Queue Manager and CHINIT
  - Keeps both Queue Manager and CHINIT statistics synchronized in time
- **Valid values for STATIME**
  - **Default from the CSQ4SYSP macro - 30 (minutes)**
  - **Zero -** use the global SMF interval
  - **Non-zero -** SMF data will be collected when the specified interval expires. The value is in minutes
- **To set a different interval dynamically**
  - Use **SET SYSTEM  STATIME** command
    - Takes effect immediately after the current interval expires

    **+cpf  SET SYSTEM STATIME(10)**

# Additional Controls for Channel Accounting

- **Queue Manager attribute: STATCHL**
  - **OFF (default value)**
    - Disables channel accounting for channels with STATCHL(QMGR)
  - **LOW/MEDIUM/HIGH**
    - All have the same effect
    - Enables channel accounting for channels with STATCHL(QMGR)
  - **NONE**
    - Disables channel accounting for all channels

# Additional Controls for Channel Accounting

- **Channel attribute: STATCHL**
  - **QMGR (default value)**
    - Channel accounting is controlled by the setting of the Queue Manager STATCHL attribute
  - **LOW/MEDIUM/HIGH**
    - All have the same effect
    - Enables channel accounting for this channel
  - **OFF**
    - Disables channel accounting for this channel

# Controlling Channel Accounting

- A new attribute called STATCHL which allows statistics collection granularity at the channel level has been added to the channel definition.

It can be specified on Sender, Receiver, Server, Requester, Cluster Sender and Cluster Receiver channels.

- STATCHL can also be specified at the Queue Manager level to allow channels to inherit a system wide setting.

- The amount of data collected is a superset of that collected on the distributed platforms with the STATCHL event message.

- The queue manager object also has a STATACLS which sets the STATCHL value for automatically defined cluster sender channels.

# Channel Accounting for auto-defined cluster channels

- **Queue Manager attribute: STATACLS**
  - **QMGR (default)**
    - Channel accounting for auto-defined cluster sender channels is controlled by the setting of the Queue Manager **STATCHL** attribute
  - **LOW/MEDIUM/HIGH**
    - Have the same effect
    - Enables channel accounting for auto-defined cluster sender channels
  - **OFF**
    - Disables channel accounting for auto-defined cluster sender channels

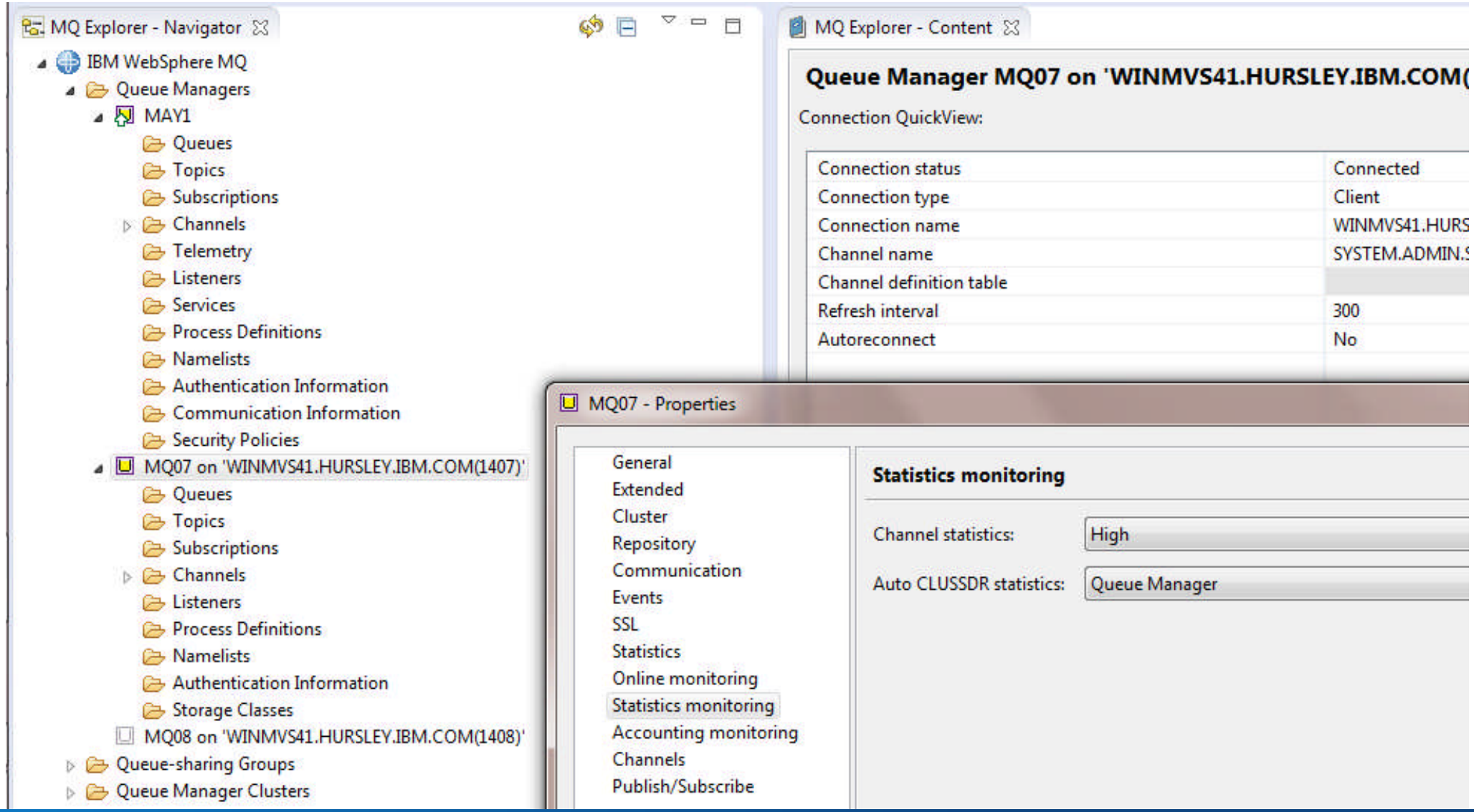
# Channel Accounting for auto-defined cluster channels

- The queue manager object also has a STATACLS which sets the STATCHL value for automatically defined cluster sender channels.

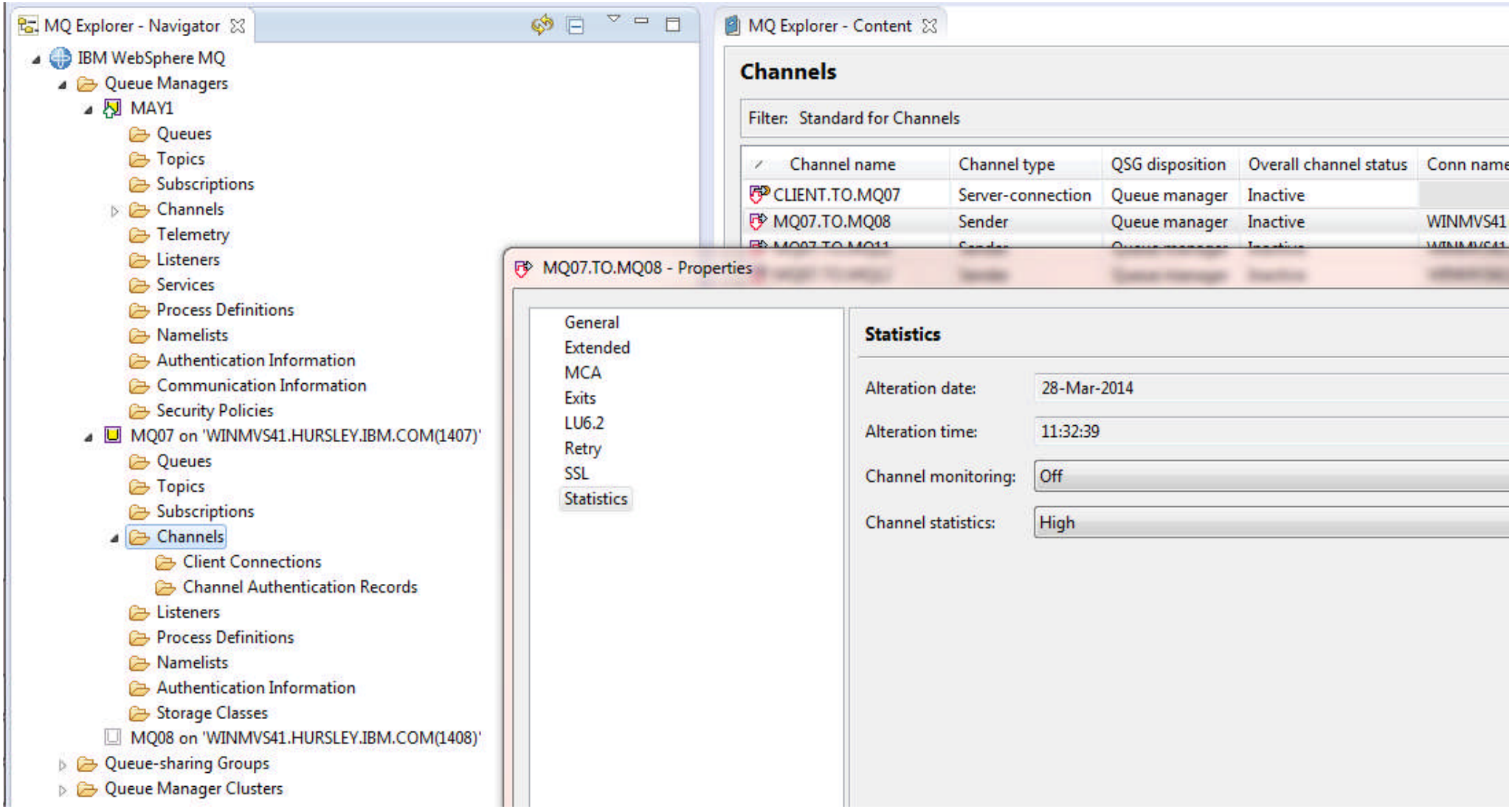
# Channel Accounting for SVRCONN channels

- **For SVRCONN channels**
  - Set **STATCHL** at the QMGR level
- **Enables it for all client connections**
- **But, be careful as channel accounting data is captured at:**
  - Each SMF statistics interval (STATIME), and
  - When a channel ends data is captured and held until next interval
  - Hence, frequent client connects/disconnects can result in a lot of data!

# MQ Explorer - Enabling Channel Statistics on QMGR



# MQ Explorer - Enabling Channel Statistics on channel



# New console messages for CHINIT SMF

- **CSQX076I**
  - Issued during CHINIT startup
  - Reports values of Queue Manager attributes STATCHL and STATACLS

```
...
22.59.05 STC13103  +CSQX074I !MQ07 CSQXGIP MONCHL=OFF, MONACLS=QMGR
22.59.05 STC13103  +CSQX075I !MQ07 CSQXGIP ADOPTMCA=ALL, ADOPTCHK=ALL
22.59.05 STC13103  +CSQX076I !MQ07 CSQXGIP STATCHL=OFF, STATACLS=QMGR
22.59.05 STC13103  +CSQX078I !MQ07 CSQXGIP IGQ=DISABLED, CHADEXIT=
22.59.05 STC13103  +CSQX079I !MQ07 CSQXGIP TRAXSTR=YES, TRAXTBL=2
...
```

# New console messages for CHINIT SMF

- **A new task, CSQXSMFT, is attached for CHINIT SMF**
- **If this task encounters an error, the following message is issued:**

  **CSQX124E csect-name SMF task ended abnormally, RC=retcode, reason=reason**

  - An abend (with a dump) is issued

- **If other errors are encountered while processing CHINIT SMF:**

  **CSQX122E** csect-name Failed to process channel accounting, RC=retcode

  **CSQX123E** csect-name Failed to process channel initiator statistics, RC=retcode

  **CSQX125I** csect-name SMF data incomplete

# New console messages for CHINIT SMF

If the MEMLIMIT parameter is not set in the channel initiator JCL, the amount of virtual storage above the bar may be set from by the MEMLIMIT parameter in the SMFPRMxx member of SYS1.PARMLIB or from the IEFUSI exit.

If the MEMLIMIT is set to restrict the above bar storage below the required level, the channel initiator will issue the **CSQX124E “SMF task ended abnormally”** message and class 4 accounting and statistics trace will not be available.

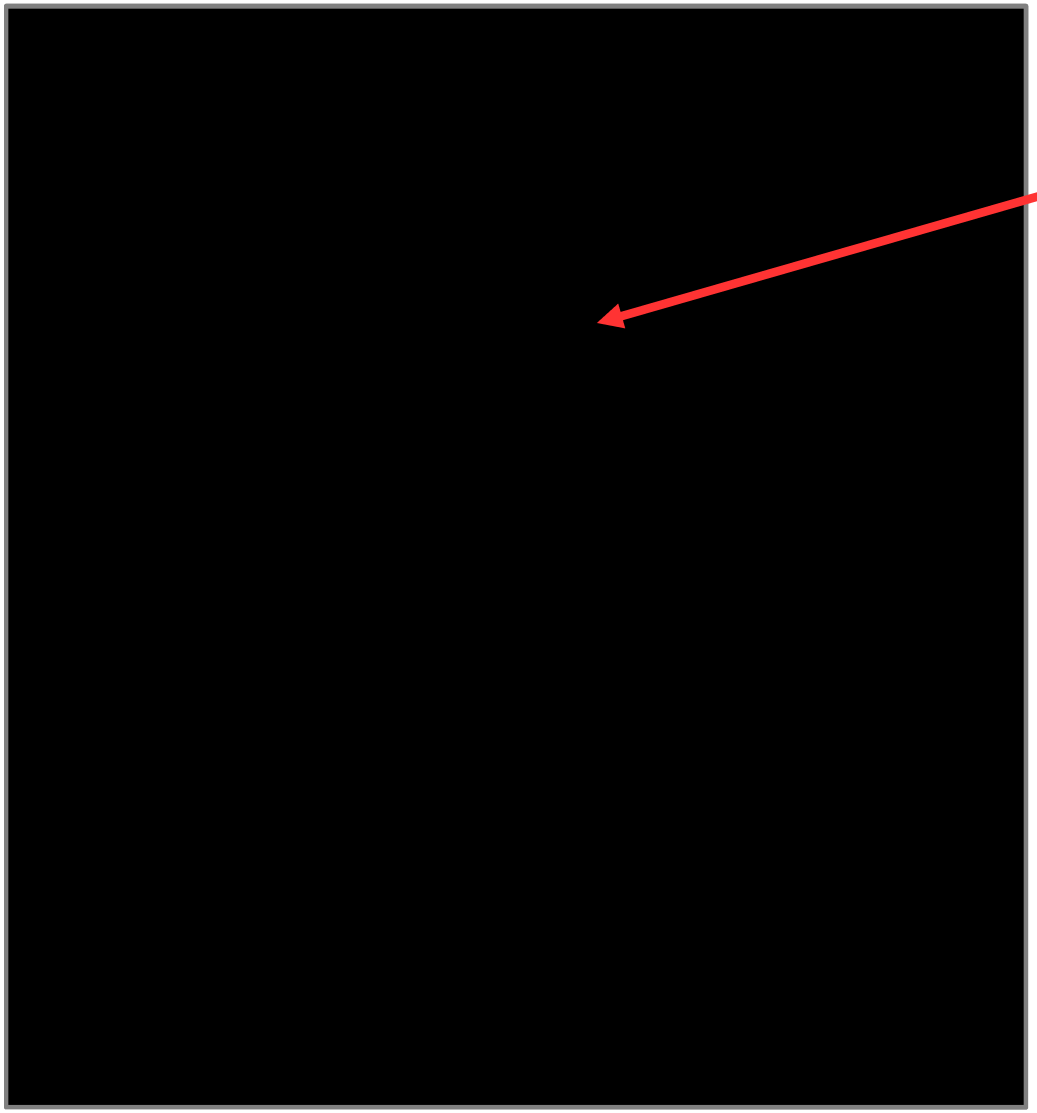
# Interpreting SMF data

- **Details of new SMF records are documented in the InfoCenter**
  - Copybooks that map the records are shipped
- **SupportPac MP1B has been updated to:**
  - Format new SMF data
  - **MQSMF** displays formatted records
    - Outputs information to various files (DDs)
    - Highlights potential out-of-line conditions
    - Can output comma-separated values (CSV) to import in spreadsheets
- **Sample program CSQ4SMFD.C (run by CSQ4SMFJ.JCL) has also been updated**
  - Formats CHINIT SMF data in a dump like fashion

# Interpreting SMF data

- WebSphere MQ provides detailed information describing the SMF records it produces. These can be used to understand the data that is generated and produce utilities to interpret this information.
- The category 2 SupportPac MP1B provides a program called MQSMF that can be used to format the SMF records instead. This program analyses SMF records and outputs information to various files (DDs) if they are specified. In addition to formatting the data in to human-readable output, it also has support for highlighting various conditions that might warrant further attention by administrators.
- MQSMF can also output data as comma separated values (CSV) that can be readily imported into spreadsheets for further analysis.

# MQSMF - Example JCL



**NEW DD cards**

```
//CHINIT        DD SYSOUT=*
//CHINCSV       DD SYSOUT=*
//CMESSAGE      DD SYSOUT=*
//ADAP        DD SYSOUT=*
//ADAPCSV       DD SYSOUT=*
//DISP          DD SYSOUT=*
//DISPCSV       DD SYSOUT=*
//DNS           DD SYSOUT=*
//DNSCSV        DD SYSOUT=*
//SSL           DD SYSOUT=*
//SSLCSV        DD SYSOUT=*
//DCHS          DD SYSOUT=*
//DCHSCSV       DD SYSOUT=*
//DCHSSUM       DD SYSOUT=*
```

# CHINIT Statistics Summary (//CHINIT)

MVCA,MQPV,2014/03/18,13:00:00,VRM:800,
From 2014/03/18,12:45:00.015222 to 2014/03/18,13:00:00.083630 duration 900.068408 seconds
Peak number used of current channels..... 4
Peak number used of active channels ..... 0
MAXCHL. Max allowed current channels.....9999
ACTCHL. Max allowed active channels.....9999
TCPCHL. Max allowed TCP/IP channels.....9999
LU62CHL. Max allowed LU62 channels..... 200
Storage used by Chinit..... 436MB

# CHINIT Statistics Summary (//CHINIT)

**The next few slides show output of the CHINIT's SMF data, this has been formatted by supportpac MP1B - other formatters are available.**

**The output is taken from one of our test systems.**

**On this slide, the CHINIT summary data produced by the //CHINIT DD card is shown.**

**This CHINIT has peaked at 4 current and the address space is using 436MB of storage.**

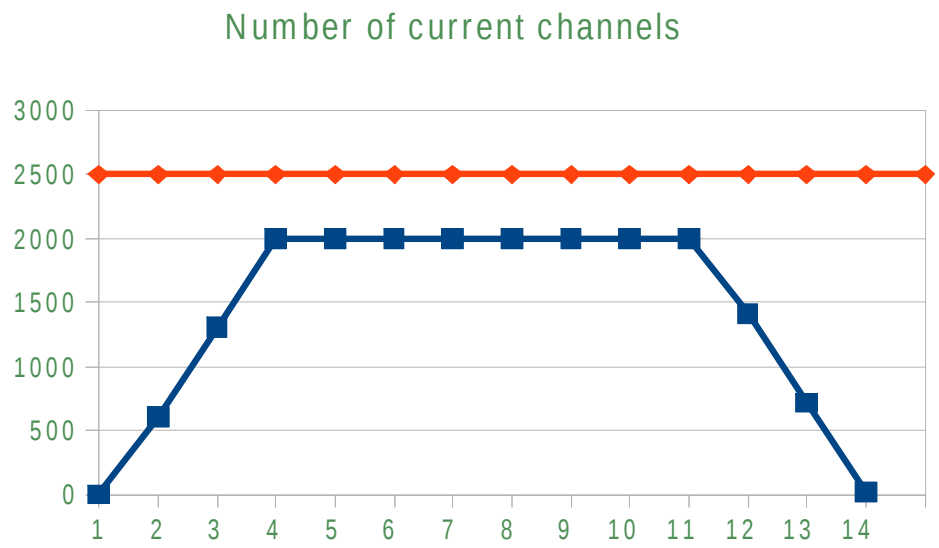
**Notes:**

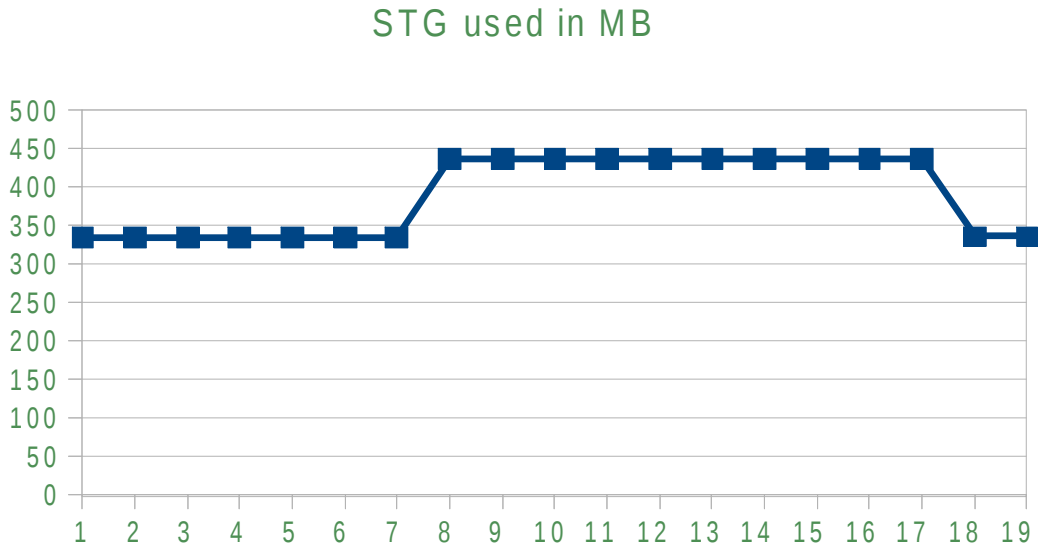
**1) A current channel is "active" unless it is in RETRYING, STOPPED, or STARTING state.**

**2) A channel is "current" if it is in any state other than inactive.**

# CHINIT Statistics Summary (//CHINITCSV)

- **Number of current and active channels**
  - How close are you getting to the maximums?
- **Channel initiator storage usage**
  - 31-bit usage – currently not much in 64-bit for the channel initiator
- **Are these trending upwards?**



# Dispatcher Task Statistics

- **Dispatcher Task Statistics are reported in the DISP and DISPCSV output of the MQSMF program**
  - The DISP file is the formatted report
  - The DISPCSV is the comma separated values version of the file

- **Note that on the display (next foil) some fields have been removed to save space.**

# Dispatcher Task Statistics

| Task, | Type, | Requests, | Busy %, | CPU used, Seconds, | CPU %, | "avg CPU", uSeconds, |
|---|---|---|---|---|---|---|
| 0, | DISP, | 26587, | 0.4, | 0.592463, | 0.1, | 22, |
| 1, | DISP, | 26963, | 0.3, | 0.588092, | 0.1, | 22, |
| 2, | DISP, | 864329, | 2.7, | 2.545668, | 0.3, | 38, |

Dispatcher task number

Dispatcher 2 is busy, other tasks are less busy as some channels against them have stopped

Dispatchers have ample capacity

4.9 secs of CPU time used by Dispatcher tasks

Average CPU for Dispatcher requests

# Dispatcher Task Statistics – Notes

The example data shows three (0-2) of the five dispatcher tasks (0 → 4) defined at the time the SMF record was cut. One dispatcher task, task 2, is processing more requests than the others. This is normal, as some channels might stop so the dispatcher is processing fewer channels. Also, some channels can be busier than others.

- 4.9 seconds of CPU were used by the dispatchers.

- Dispatcher requests are generally TCP send and receive requests and channel exit requests. The average request used 38 microseconds of CPU. Elapsed time is alos reported but not shown on these foils.

- This report also shows the average time per request. The average CPU used per request depends on the message traffic, for example, bigger messages use more CPU than smaller messages.

- The %Busy indicates if a dispatcher has spare capacity so this report would help an MQ administrator work out if there are enough dispatchers.

# Dispatcher Task Statistics - Continued

- The next section in the report shows the number of channels per dispatcher.

```
   0,DISP, number of channels on this TCB,    3
   1,DISP, number of channels on this TCB,    2
   2,DISP, number of channels on this TCB,   15
   3,DISP, number of channels on this TCB,    0
   4,DISP, number of channels on this TCB,    0
Summ,DISP, number of channels on all TCBs,   20
```

- As expected, dispatcher 2 shows more channels on the TCB during this interval.

# Adapter Task Statistics

```
MV45,MQ20,2014/04/08,20:43:57,VRM:800,
From 2014/04/08,20:41:54.984681 to 2014/04/08,20:43:57.237939
duration 122.253258 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
   ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,ADAP,   127599,   16.5,  0.953615,   0.8,         7,       158
   1,ADAP,    46790,    7.6,  0.309678,   0.3,         7,       199
   2,ADAP,    13702,    3.2,  0.065380,   0.1,         5,       284
   3,ADAP,     2909,    0.7,  0.029541,   0.0,        10,       279
   4,ADAP,      395,    0.1,  0.003179,   0.0,         8,       392
   5,ADAP,       37,    0.0,  0.000241,   0.0,         7,       149
   6,ADAP,       10,    0.0,  0.000175,   0.0,        17,       111
   7,ADAP,        0,    0.0,  0.000000,   0.0,         0,         0
Summ,ADAP, 191442,    3.5,  1.361809,   0.1,         7,       179
```

MQI requests are processed by first free adapter so adapters lower in the list process fewer requests

Difference could indicate wait for I/O due to commit or disk read

# Adapter Task Statistics (//ADAP + //ADAPCSV)

**This shows an example of the adapter task statistics report.**

**The adapters process MQI requests. Each MQI request uses the first free adapter so expect to see decreasing busyness.**

**Some of these requests might wait, for example, for log I/O during a commit, so the difference between the average CPU time and average Elapsed Time per request can be quite large.**

**This is the report that an MQ administrator would use to ensure that there are enough adapter tasks defined. A channel should not generally need to wait for an adapter.**

**In this example, we never used all the adapters. So, there is no need to add more adapters. If the last adapter is very busy, consider increasing the number of adapter tasks.**

# DNS Task Statistics

```
MV45,MQ20,2014/04/08,20:41:54,VRM:800,
From 2014/04/08,20:40:07.101220 to 2014/04/08,20:41:54.984681 duration
107.883460 seconds
Task,Type,Requests,Busy %,      CPU used, CPU %,
    ,    ,        ,       ,       Seconds,      ,
   0,DNS ,        24,   0.0,     0.007980,  0.0,
Summ,DNS ,        24,   0.0,     0.007980,  0.0,



"avg CPU","avg ET",longest ,date       ,time
 uSeconds,uSeconds,uSeconds,            ,
      332,     1031,    24284,2014/04/08,20:41:49.573730
 Summ,332,     1031,    24284,2014/04/08,20:41:49.573730
```

Only 1 DNS task, not busy

Longest DNS resolution request

# DNS Task Statistics

- **There is only one DNS task**
  - If this task is very busy, let IBM know!
- **Longest request was 24284 microseconds**
- **Date and time fields show when this happened**
- **Message CSQX788I issued if DNS lookup takes >3 secs**

```
CSQX788I csect-name DNS lookup for address address using function 'func'
  took n seconds
```

# SSL Task Statistics

```
MV45,SS09,2014/04/10,23:22:24,VRM:800,
From 2014/04/10,22:53:26.883960 to 2014/04/10,23:22:24.204176 duration
1737.320215 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
    ,   ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,SSL ,  109843,   0.3,     0.594580,  0.0,         5,      42,
   1,SSL ,  130180,   0.3,     0.713966,  0.0,         5,      41,
   2,SSL ,  117544,   0.3,     0.703146,  0.0,         6,      42,
   3,SSL ,  145944,   0.4,     0.830535,  0.0,         6,      43,
   4,SSL ,  123825,   0.3,     0.679656,  0.0,         5,      43,

longest ,date      ,time
uSeconds,          ,
  229638,2014/04/10,22:54:34.264949
  255082,2014/04/10,22:54:54.302855
  230501,2014/04/10,22:54:43.958105
  280241,2014/04/10,22:54:53.499979
  361212,2014/04/10,22:54:53.599940
```

Low average CPU time with higher elapsed time may be due to cryptographic off-load to card

Longest busy times due to lots of channels starting together

# SSL Task Statistics

- CPU time expected to be less than elapsed time because cryptographic operations are offloaded
- The long busy times seen in the example were due to lots of channels starting up at the same time
- Adding more SSL tasks might not improve performance if waiting for external hardware, such as a single cryptographic card

# Channel Accounting Data – Sender Channel Part 1

| | | | |
|---|---|---|---|
| 127.0.0.1 | MQ89_1 | Connection name | 127.0.0.1 |
| 127.0.0.1 | MQ89_1 | Remote qmgr/app | MQ89 |
| 127.0.0.1 | MQ89_1 | Channel disp | PRIVATE |
| 127.0.0.1 | MQ89_1 | Channel type | SENDER |
| 127.0.0.1 | MQ89_1 | Channel status | RUNNING |
| 127.0.0.1 | MQ89_1 | Channel STATCHL | HIGH |
| 127.0.0.1 | MQ89_1 | Channel started date & time | 2014/04/08,19:41:48 |
| 127.0.0.1 | MQ89_1 | Channel stopped time | |
| 127.0.0.1 | MQ89_1 | Channel status collect time | 2014/04/08,19:43:57 |
| 127.0.0.1 | MQ89_1 | Last msg time | 2014/04/08,19:43:52 |
| 127.0.0.1 | MQ89_1 | Active for | 122 seconds |
| 127.0.0.1 | MQ89_1 | Batch size | 50 |
| 127.0.0.1 | MQ89_1 | Messages/batch | 38.9 |
| 127.0.0.1 | MQ89_1 | Number of messages | 2,998 |
| 127.0.0.1 | MQ89_1 | Number of persistent messages | 1,506 |
| 127.0.0.1 | MQ89_1 | Number of batches | 77 |
| 127.0.0.1 | MQ89_1 | Number of full batches | 42 |
| 127.0.0.1 | MQ89_1 | Number of partial batches | 35 |
| 127.0.0.1 | MQ89_1 | Buffers sent | 3,319 |
| 127.0.0.1 | MQ89_1 | Buffers received | 109 |
| 127.0.0.1 | MQ89_1 | Xmitq empty count | 13 |

# Channel Accounting Data – Sender Channel - Notes

- **Channel accounting data is reported in the DCHS and DCHCVS output file from the MQSMF program**
- **Each record identifies each channel with its connection name, channel name and remote queue manager name (for queue manager to queue manager channels). For SVRCONN channels the remote application name is given.**
- **Some of the batches were not full. Target batch size was 50 but average achieved batch size was 38.9. The number of full and partial batches are shown.**
  - `BATCHSZ`, `BATCHLIM` and message arrival impacts this
- **About half the messages sent were persistent**

# Channel Accounting Data – Sender Channel Part 2

| | | | | |
|---|---|---|---|---|
| 127.0.0.1 | MQ89_1 | Message data | 17,198,653 | 16 MB |
| 127.0.0.1 | MQ89_1 | Persistent message data | 4,251,780 | 4 MB |
| 127.0.0.1 | MQ89_1 | Non persistent message data | 12,946,873 | 12 MB |
| 127.0.0.1 | MQ89_1 | Total bytes sent | 17,200,221 | 16 MB |
| 127.0.0.1 | MQ89_1 | Total bytes received | 3,052 | 2 KB |
| 127.0.0.1 | MQ89_1 | Bytes received/Batch | 39 | 39 B |
| 127.0.0.1 | MQ89_1 | Bytes sent/Batch | 223,379 | 218 KB |
| 127.0.0.1 | MQ89_1 | Batches/Second | 0 | |
| 127.0.0.1 | MQ89_1 | Bytes received/message | 1 | 1 B |
| 127.0.0.1 | MQ89_1 | Bytes sent/message | 5,737 | 5 KB |
| 127.0.0.1 | MQ89_1 | Bytes received/second | 25 | 25 B/sec |
| 127.0.0.1 | MQ89_1 | Bytes sent/second | 140,985 | 137 KB/sec |
| 127.0.0.1 | MQ89_1 | Compression rate | 0 | |
| 127.0.0.1 | MQ89_1 | Exit time average | 0 | uSec |
| 127.0.0.1 | MQ89_1 | DNS resolution time | 0 | uSec |
| 127.0.0.1 | MQ89_1 | Net time average | 312 | uSec |
| 127.0.0.1 | MQ89_1 | Net time min | 43 | uSec |
| 127.0.0.1 | MQ89_1 | Net time max | 4,998 | uSec |
| 127.0.0.1 | MQ89_1 | Net time max date&time | 2014/04/08,19:43:52 | |

# Channel Accounting Data – Sender Channel Part 2

- **Total message data of about 16MB sent during the interval**
- **The average number of bytes sent per message was about 5KB**
- **Bytes sent/received per second is:**
  - Average/interval
- **As this is a sender type channel, as expected, the bytes sent is greater than the bytes received**
- **Monitor channel usage over time to look for trends**

# Channel Accounting Summary

```
MVS,MQ,date,time,VRM,channelType,count,Persistent,NonPersistent,'P/Sec','NP/Sec'
MVCA,MQPV,2014/06/30,11:30:00,VRM:800,RECEIVER,2,75720,0,3786,0
MVCA,MQPV,2014/06/30,11:30:00,VRM:800,total,2,75720,0,3786,0
MVCA,MQPH,2014/06/30,11:30:00,VRM:800,SENDER,2,75720,0,2611,0
MVCA,MQPH,2014/06/30,11:30:00,VRM:800,total,2,75720,0,2611,0
MVCA,MQPH,2014/06/30,11:34:04,VRM:800,SENDER,23,86237508,0,559983,0
MVCA,MQPH,2014/06/30,11:34:04,VRM:800,total,23,86237508,0,559983,0
```

Sender channel activity
Shown over 2 intervals

These are the number of persistent and nonpersistent messages sent during the intervals. In this example, all were persistent.

# Channel Accounting Summary

- This information is from the DCHSSUM output from the MQSMF program

# CHINIT Messages

- **Some Examples:**

**MQCHIN001W The high water mark of the number of active channels >50 % of max channels**

**MQCHIN007I Dispatcher task is nn% busy on average**

**MQCHIN008I Adapter task is nn% busy on average**

**MQCHIN009I SSL task is nn% busy on average**

- **There are more examples in the documentation for SupportPac MP1B**

# Overhead for statistics and accounting

- **An MQ V8 Channel Initiator allocates approximately 190MB of above the bar virtual storage for Channel Initiator Statistics and Channel Accounting Data, regardless of whether CLASS(4) trace is enabled.**
- **Recommend Channel Initiator is allowed access to a minimum of 256MB of virtual storage i.e. set MEMLIMIT=256M if CLASS(4) trace is enabled.**
- **Release specific Performance Support Pack MP1J (due out soon)**
  - Indicates **1-2% CPU overhead** for collecting CHINIT statistics and Channel accounting data

# Overhead for statistics and accounting

If the MEMLIMIT parameter is not set in the channel initiator JCL, the amount of virtual storage above the bar may be set from by the MEMLIMIT parameter in the SMFPRMxx member of SYS1.PARMLIB or from the IEFUSI exit.

If the MEMLIMIT is set to restrict the above bar storage below the required level, the channel initiator will issue the CSQX124E “SMF task ended abnormally” message and class 4 accounting and statistics trace will not be available.

# Notes on High CPU in CHINIT Address space

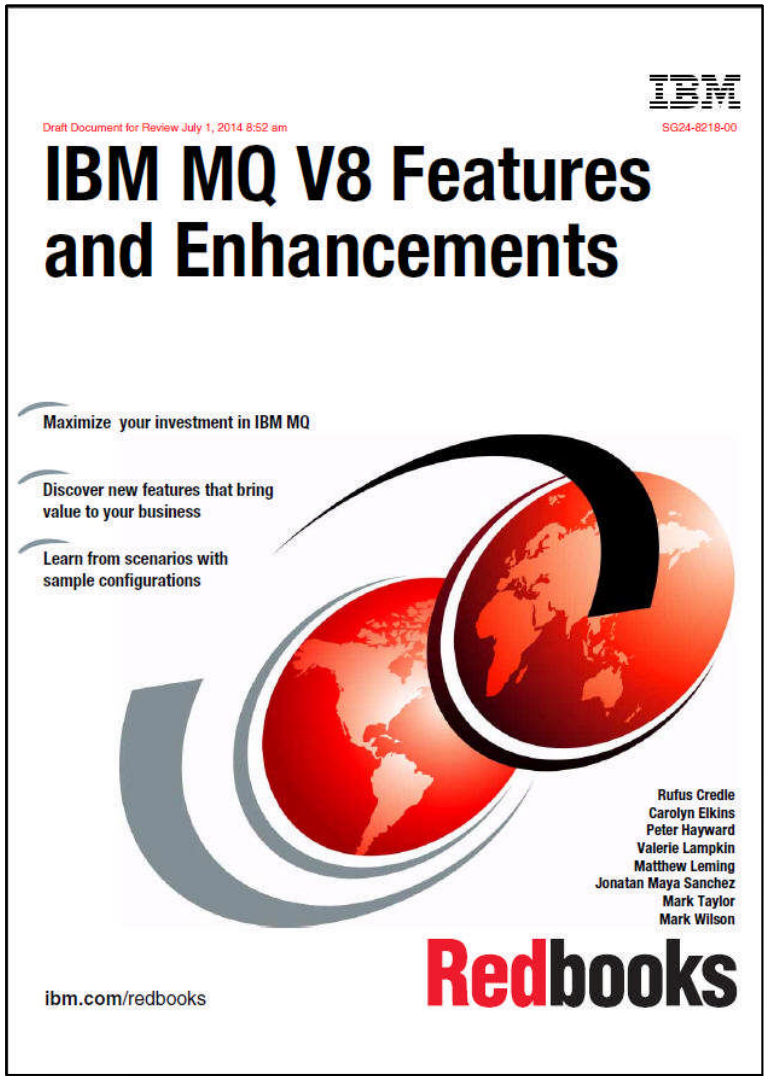
- **Level 2 has a report on some things to examine that can be found here:**

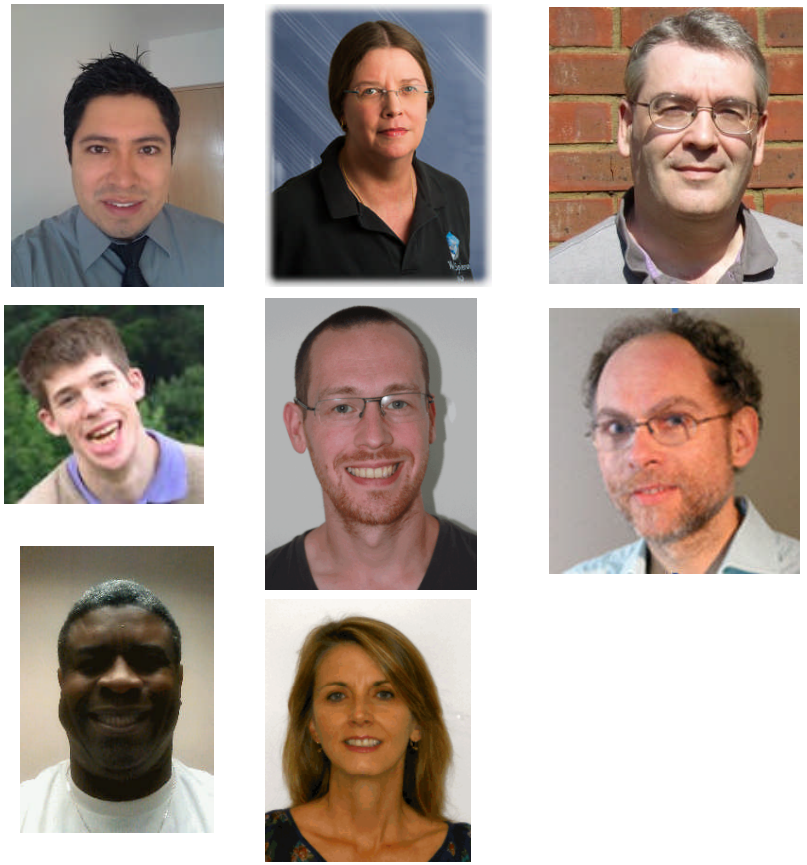
  **http://www-01.ibm.com/support/docview.wss?uid=swg27010914&aid=1**

- **Check the CURDEPTH of your SYSTEM.CHANNEL.SYNCQ**
  - If >1000, check that the Queue has INDEXTYPE(MSGID) set
- **Check that you have enough Adapter tasks**
  - See performance tuning recommendations for the CHINIT in Performance SupportPac MP16
- **Check your MAXCHL parameter as this can influence the distribution of channels to dispatchers = See MP16**
- **Check the number of dispatchers you have defined**
  - The first ( MIN( (MAXCHL / CHIDISPS ) , 10 ) channels to start are associated with the first dispatcher TCB and so on until all dispatcher TCBs are in use. The effect of this for small numbers of channels and a large MAXCHL is that channels are NOT evenly distributed across dispatchers.

    We suggest setting MAXCHL to the number of channels actually to be used where this is a small fixed number.
  - We suggest CHIDISPS(20) for systems with more than 100 channels. We have seen no significant disadvantage in having CHIDISPS(20) where this is more dispatcher TCBs than necessary.
  - See MP16

# And … already available



https://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg248218.html

# New Redbook covers MQ V8

This redbook covers the new features introduced in V8 that we have just discussed. The book is currently available in draft form. The final version is expected to be made available soon.