IBM Software Group

WebSphere MQ Problem Diagnostics
Reading Traces

Open Mic

24 March 2010

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS™

Agenda

- Introduce the panel of experts
- IBM® WebSphere® MQ trace concepts and emailed questions
  - Steps for collecting trace
  - Common issues collecting trace
  - Performance effects of trace
  - Tools for examining trace
  - Explanation of the trace fields
  - Debugging applications with trace
- Open telephone lines for questions
- Summary

Panel of Experts

Panelist	Role at IBM
Chris Andrews	WebSphere MQ JMS Level 3 Software Developer
Rick Armstrong	WebSphere MQ for z/OS Level 2 Customer Support
Douglas Burns	WebSphere MQ for z/OS Level 3 Software Developer
Damon Cross	WebSphere MQ for z/OS Level 3 Software Developer
Jason Edmeades	WebSphere MQ Service Architect
Rhys Francis	WebSphere MQ for z/OS Level 3 Software Developer
Justin Fries	WebSphere MQ and SIB Level 2 Customer Support
Lauranette Wheeler	WebSphere MQ for z/OS Level 2 Customer Support

Introduction

- The IBM Support Center frequently requests traces to analyze WebSphere MQ behavior and to diagnose problems
- We will explain how to collect WebSphere MQ trace effectively for IBM Support and for examining your own applications
- We will discuss WebSphere MQ trace for distributed platforms: AIX®, HP-UX®, Linux®, Solaris®, and Windows®; for z/OS®; and for applications using the Java™ programming language
- You will have the opportunity to ask questions about trace and how you can solve application problems using trace

Question 1

- How do I collect WebSphere MQ trace?

Question 1 – Distributed Platform Trace Options

■ Use the start trace program	`strmqtrc`
■ Choose the scope of the trace	
▸ Trace everything (default)	`-e`
▸ Trace a queue manager	`-m MY.QMGR`
▸ Trace matching queue managers	`-m “*.TEST”`
▸ Trace programs by name	`-p “runmq*,amqsput”`
▸ Trace a process identifier	`-i 1701`
▸ Trace a thread in a process	`-i 1701.13`
■ Choose the type of trace points	
▸ Trace all points (default)	`-t all`
▸ Trace all points with high detail	`-t all -t detail`
▸ Trace application API calls	`-t api`

Question 1 – Distributed Platform Trace Options

- Choose the amount of data to include
  - Trace no message data `-d 0`
  - Trace some message data `-d 4096`
  - Trace complete message data `-d all`
  - Trace the start and end (default) `-d -2`

- Choose a trace file size limit in MB `-l 100`
  - Every trace file will roll over when it reaches this size
  - Processes can have two traces
    - The current trace file `AMQ1701.0.TRC`
    - The previous trace file `AMQ1701.0.TRS`
  - This is called a wrapping trace

Question 1 – Distributed Platform Trace Options

- Choose a trigger to begin the trace
  - Valid triggers are FDC Probe Ids `-b FDC=HL142100`
  - Lists and wildcards are allowed `-b “FDC=KN247001,ZX*”`

- Choose a trigger to conclude the trace
  - Use Probe Id lists and wildcards `-c “FDC=XC130*”`
  - Or stop after several seconds `-c interval=60`
  - Use one or the other

- Multiple traces can run simultaneously
  - Up to fifteen can be set at once
  - To show what traces are set `strmqtrc -s`

Question 1 – Distributed Platform Trace Options

- Use the stop trace program
  - Stop <u>all</u> active traces
  - To stop a specific trace, match its original options precisely

```
endmqtrc
-a
-p “runmq*,amqsput”
```

- Use the format trace program
  - To print a file <u>summary</u> only
  - Give the file names to format
  - TRC and TRS files are merged sensibly for each process

```
dspmqtrc
-s
AMQ*
```

- The format trace program is not required on Windows

Question 1 – JMS Clients

- Enable trace by setting this property

```
com.ibm.msg.client.commonservices.trace.status=ON
```

- Configure the output directory using this property

```
com.ibm.msg.client.commonservices.trace.outputName=<directory>
```

- For example, from the command line run:

```
java -Dcom.ibm.msg.client.commonservices.trace.status=ON
     -Dcom.ibm.msg.client.commonservices.trace.outputName=/home/myhome
      MyJmsApplication
```

Question 1 – JMS Client Options

- Selecting specific packages to trace using properties
  - IBM Support will advise on package names

```
com.ibm.msg.client.commonservices.trace.include=<package name>
com.ibm.msg.client.commonservices.trace.exclude=<package name>
```

- Limiting trace file size and number of trace files

```
com.ibm.msg.client.commonservices.trace.limit=<size in bytes>
com.ibm.msg.client.commonservices.trace.count=<number of files>
```

- Create compressed trace files

```
com.ibm.msg.client.commonservices.trace.compress=TRUE
```

Question 1 – JMS Clients in WebSphere

- Trace is integrated with the logging system available in WebSphere Application Server

- Enable trace in the Administrative Console

```
Troubleshooting
 ► Logs and Trace
    ► <choose server>
       ► Change Log Detail Levels
```

  - Configure the size and number of trace files
  - Add the following trace hooks to the trace string:

```
*=info:JMSApi=all:JMSServer=all:Messaging=all:
JMS_WASTraceAdapter=all:com.ibm.mq.*=all:jmsApi=all
```

Question 1 – Base Java Clients

- Create a text file to hold the trace properties

```
C:\TEMP\trace.properties
```

- Use these property values as a template for the file

```
Diagnostics.MQ=enabled
Diagnostics.Java=all
Diagnostics.Java.Trace.Detail=high
Diagnostics.Java.Trace.Destination.File=enabled
Diagnostics.Java.Trace.Destination.Console=disabled
Diagnostics.Java.Trace.Destination.Pathname=.
```

- Start the base classes for Java application as follows:

```
java -Dcom.ibm.mq.commonservices=C:\TEMP\trace.properties MyApp.class
```

Question 1 – z/OS Options and Example

- The following options are available to start a trace

```
START TRACE(GLOBAL) DEST(GTF) CLASS(*) RMID(*)
            ACCTG        RES
            CHINIT       SMF
            STAT         SRV
```

- For example, to start and later stop channel initiator trace

```
ALTER QMGR TRAXTBL(500)
START TRACE(CHINIT)

STOP TRACE(CHINIT)
```

Question 1 – z/OS Example

- Display the current settings before changing trace options

```
DISPLAY TRACE
```

- Stop the current trace, then start with the new options

```
STOP TRACE(*)
SET SYSTEM TRACTBL(500)
START TRACE(GLOBAL) DEST(RES) CLASS(*) RMID(*)
```

- Restart the trace with its original settings, for example

```
STOP TRACE(*)
START TRACE(GLOBAL) DEST(GTF) CLASS(*) RMID(*)
```

Question 1 – z/OS GTF Trace

- The following command will start a GTF trace

xx	The name of the GTF procedure to use
yy	An identifier for this occurrence of trace
ssid	The name of the queue manager
appl	The name of the application making MQI calls

```
/START GTFxx.yy
 *01 AHL100A SPECIFY TRACE OPTIONS
  R 01,TRACE=JOBNAMEP,USRP
 *02 AHL101A SPECIFY TRACE EVENT KEYWORDS - JOBNAME=,USR=
  R 02,JOBNAME=(ssidMSTR,ssidCHIN,appl),USR=(5E9,5EA,5EB,5EE,F6C)
 *03 AHL102A CONTINUE TRACE DEFITION OR REPLY END
  R 03,END
 *04 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
  R 04,U
```

- This command stops the GTF trace using its identifier

```
/STOP yy
```

Question 2

- What are the most common issues when collecting trace?

Question 2 – Distributed Platforms and Java

- Trace files grow too large over time
  - Limit the maximum trace file size properly
  - Use a separate filesystem for trace files
- Trace affects performance too much
  - Limit the trace scope and trace type (or packages)
- Trace is hard to gather for intermittent problems
  - Use a wrapping trace with a stop trigger
- Trace did not capture the problem
  - Verify the time frame covered by the trace
  - Increase the size of the trace to limit wrapping
  - Confirm error messages and FDCs appear in the trace

Question 2 – z/OS

- No GTF trace was collected
  - Ensure the trace task started: Look at `AHL` messages
  - Use the correct MSTR and application job names
  - The queue manager trace must have `DEST(GTF)`
- The trace will not format
  - IPCS cannot access the MQ formatter in `SCSQAUTH`
  - Bad dataset attributes: Invalid allocation or altered in transfer
- Trace did not capture the problem
  - The dataset is too small: GTF uses primary allocation only
  - Limit trace to required areas, for example `USR=(5E9,5EA)`
  - “LOST EVENTS”: Increase GTF collection job priority
  - Stop trace promptly in order to prevent wrapping

Question 3

- What is the performance impact of trace?

Question 3 – Distributed Platforms and Java

- Turning on trace hits performance...
  ...but it may be the only way to get information!

- IBM does not publish official numers for running with trace

- Worst case costs (single disk, I/O bound)
  - AIX shows about a 32x slowdown
  - Windows shows about a 38x slowdown
    - Slower due to writing larger, formatted output
  - Java and JMS trace shows about a 10x slowdown

Question 3 – z/OS

- Refer to SupportPac™ MP16 for capacity planning and tuning
  - Global trace
    - An internal trace, `TRACE(GLOBAL) DEST(RES)` increases CPU costs by approximately 30-90%
    - This is based on the amount the ASID is using
  - Accounting trace
    - `TRACE(ACCTG) CLASS(1)` costs an extra 2-3%
    - `TRACE(ACCTG) CLASS(3)` costs an extra 5-10%
  - Channel initiator trace
    - `TRACE(CHINIT)` increases SDR/RCVR costs by 25-55%
  - Statistics trace
    - TRACE(STAT) costs are insignificant

Question 4

- Does IBM have any scripts or tools that I can run to go through the accumulated traces?

Question 4 – z/OS GTF Filters

- Filter GTF trace in IPCS either interactively (TSO) or in batch
  - `GTF USR(5E9)` gives API entry only
  - `GTF USR(5EA)` gives API exit only
  - `GTF USR(5E9,5EA)` gives API entry and exit

- Use the `JOBNAME` parameter to get trace from only the specified jobs, for example:
  - `GTF USR(5E9,5EA) JOBNAME(ACICSJOB1)`

- Wildcards are allowed in the `JOBNAME`: `*`, `%`

Question 4 – z/OS GTF Filters

- Filter trace based on a specific time frame
  - `STARTLOC(ddd,hh.mm.ss)`
  - `STOPLOC(ddd,hm.mm.ss)`

- The `ddd` field is the Julian day
- The `hh.mm.ss` field is the local time
- You can use `START/STOP` to specify the time in GMT

- For example:

```
GTF USR(5E9) STARTLOC(022,15.31.13)
  STOPLOC(022,15.31.22)
```

Question 5

- What do the columns and parameters mean in WebSphere MQ trace?

Question 5 – Distributed Platforms

```
   A         B           C          D                      E
00010B73 11:15:10.332807  8556.1    RSESS:000001 -----{  xcsCreateThreadMutexSem
00010B74 11:15:10.332818  8556.1    RSESS:000001 -----}  xcsCreateThreadMutexSem (rc=OK)
00012014 11:15:11.258547  8556.1    RSESS:000001 ---------}! recv (rc=Unknown(BC))
000123FB 11:15:11.281188  8556.1    RSESS:000001  0x0011FD0C 4F 44 20 20 01 00 00 00 01 ...
```

- `A:` Windows only – Cross process hex counter (1, 2, 3...)
- `B:` A timestamp in the local time zone
- `C:` `ProcessId.ThreadId` – MQ thread Id, not OS thread Id
- `D:` A new identifier in v7 for tracking activity across threads
- `E:` The trace point information
  - `{` Entry into a function: Dashes show stack depth
  - `}` Exit from a function with OK return code
  - `}!` Exit from a function with nonzero return code
  - Otherwise the line shows text or a hex data dump

Question 5 – JMS

```
       A           B       C                   D
22:46:34.935.04  0001  @5ca25ca2 c.i.m.j.remote.internal.RemoteTCPConnection(RemoteTCPConnection)
----+----+----  {  send(byte [ ],int,int) <len=32764 bytes> [0] [244] <null> [false] [1] [2]
       E               F
```

- A: Timestamp
- B: Thread identifier
- C: Object reference
- D: Class and method invocation name
- E: Stack depth and function indicator
  - { Entry to a function
  - } Exit from a function
- F: The rest of the line is the trace point data

Question 5 – z/OS

```
       A    B                                    C
ABC   USRD9 5E9 ASCB 00FADA00            JOBN LAURANEG
  D   CSQW072I ENTRY: MQ user parameter trace
  E   CLOSE
        Thread... 00000000  Userid... 00000000  00000000            Hobj..... 00000001
        RSV2..... 00000000  RSV3..... 00000000  QRPLX.... 00000000
  F                 GMT-03/18/2010 03:40:21.612397  LOC-03/18/2010 03:40:21.612397
ABC   USRD9 5EA ASCB 00FADA00            JOBN LAURANEG
  D   CSQW073I EXIT:  MQ user parameter trace
  E   CLOSE
        Thread... 00000000  Userid... 00000000  00000000            Hobj..... 00000000
        RSV2..... 00000000  RSV3..... 00000000
        CompCode. 00000000  Reason... 00000000  QRPLX.... 00000000
  F                 GMT-03/18/2010 03:40:21.612493  LOC-03/18/2010 03:40:21.612493
```

- A: Trace record begins
- B: GTF record type
- C: Job name

- D: Entry to API – CSQW072I
  Exit from API – CSQW073I
- E: API call and its parameters
- F: GMT and local date and time

Question 5 – z/OS GTF API Equivalents

- The API calls shown in GTF trace differ from MQI call names

MQOPEN	OPEN
MQCLOSE	CLOSE
MQGET	GETMSG
MQPUT	PUTMSG
MQPUT1	PUTONE
MQCMIT	COMMIT
MQBACK	BACKOUT
MQINQ	INQ
MQSET	SET

- The MQCONN and MQDISC calls do not appear in GTF trace

Question 6

- How can I debug my application using WebSphere MQ trace?

Question 6 – Distributed Platform API Trace

- Collect an `api` type trace while running the application
  - `strmqtrc -t api`, recreate the issue, `endmqtrc`
- Format the traces and search for the application name
  - Or run `dspmqtrc -s` to summarize the traces
- In WebSphere MQ v7 you can trace just the application
  - `strmqtrc -t api -p TheApp`
- Look in the trace for API calls to see all parameters:

```
__________            __________
MQOPEN >>             MQOPEN <<
```

Question 6 – Distributed Platform API Example

- On a Windows system:

```
C:\mqm70\trace> strmqtrc -t api
C:\mqm70\trace> amqsput QL QM
Sample AMQSPUT0 start
target queue is QL
Hello, world.
Sample AMQSPUT0 end
C:\mqm70\trace> findstr /I amqsput *.TRC
AMQ2436.0.TRC:Process :  C:\mqm70\bin\amqsput.exe (32-bit)
```

- On a UNIX or Linux system:

```
$ strmqtrc -t api
$ amqsget QL QM
Sample AMQSGET0 start
message <Hello, world.>
no more messages
Sample AMQSGET0 end
$ dspmqtrc *.TRC
$ grep amqsget *.FMT
AMQ2891.0.FMT: 10:58:42.902930  2891.1  PID : 2891  Process : amqsget
```

Question 6 – Distributed Platform API Sample

Call

```
!! -
!! - MQCONN >>
!! - Name:
Data:-
 0x0011F820 51 4D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : QM..............
 0x0011F830 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................
 0x0011F840 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................
```

Queue manager name

Return

```
!! -
!! - MQCONN <<
!! - Name            : Input  Parm
!! - Hconn:
Data:-
 0x0011FB8C 08 C6 93 00                                      : .Æ“.
!! - Compcode:
Data:-
 0x0011FB84 00 00 00 00                                      : ....
!! - Reason:
Data:-
 0x0011FB94 00 00 00 00                                      : ....
```

Success!

Call

```
!! -
!! - MQOPEN >>
!! - Hconn:
Data:-
 0x0011FAE0 08 C6 93 00                                      : .Æ“.
!! - Objdesc:
Data:-
 0x0011F898 4F 44 20 20 01 00 00 00 01 00 00 00 51 4C 00 00 : OD  ........QL..
 0x0011F8A8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : ................
```

Queue name

Question 6 – Distributed Platform FDCs & Errors

- Finding an FDC in WebSphere MQ traces
  - The contents of the FDC are traced in full

```
10:24:14.192997  16064.1  !! - +----------------------------------------------------------+
10:24:14.193012  16064.1  !! - | WebSphere MQ First Failure Symptom Report                |
10:24:14.193019  16064.1  !! - | =========================================                |
   ...                    !! - |                                                          |
10:24:14.193096  16064.1  !! - | Probe Id          :- XC034255                            |
```

- Finding an error log message in the trace
  - The trace will show the message number and inserts

```
10:23:59.815721  16064.1  :      msgid:00007229 a1:00000009 a2:00000000 c1:QM c2:(null)
```

  - The corresponding message in `AMQERR01.LOG`

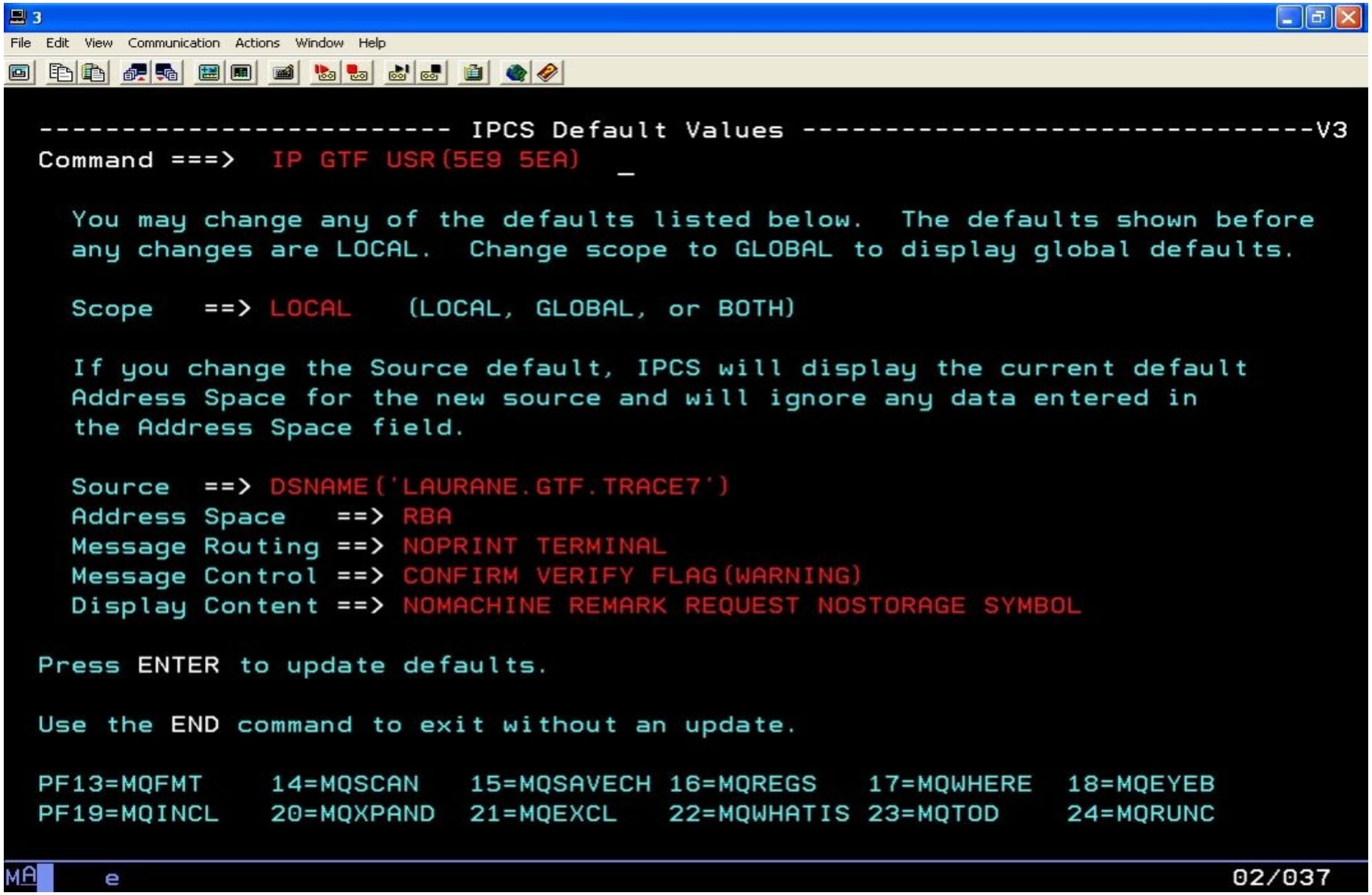
```
-----------------------------------------------------------------------------
XX/YY/2010 10:23:59 - Process(16064.1) User(XXXXXX) Program(amqzxma0.exe)

AMQ7229: 9 log records accessed on queue manager 'QM' during the log replay
phase.
```

Question 6 – JMS

- Follow the linked exceptions to the first exception:

```
com.ibm.msg.client.jms.DetailedIllegalStateException: JMSWMQ0018: Failed to connect to queue
manager 'myQM' with connection mode 'Client' and host name 'myhost.mydomain(1414)'. Check the
queue manager is started and if running in client mode, check there is a listener running.
Please see the linked exception for more information.
  at com.ibm.msg.client.wmq.common.internal.Reason.reasonToException(Reason.java:496)
  ...
  at GenericJMSClient.main(GenericJMSClient.java:38)
Caused by: com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with compcode '2'
('MQCC_FAILED') reason '2059' ('MQRC_Q_MGR_NOT_AVAILABLE').
  at com.ibm.msg.client.wmq.common.internal.Reason.createException(Reason.java:223)
  ... 7 more
Caused by: com.ibm.mq.jmqi.JmqiException: CC=2;RC=2059;AMQ9204: Connection to host
'myhost.mydomain(1414)' rejected. [1=com.ibm.mq.jmqi.JmqiException[CC=2;RC=2059;AMQ9213:
A communications error for  occurred. [1=java.net.ConnectException[Connection refused],
3=myhost.mydomain]],3=myhost.mydomain(1414),5=RemoteTCPConnection.connnectUsingLocalAddress]
  at com.ibm.mq.jmqi.remote.internal.RemoteFAP.jmqiConnect(RemoteFAP.java:1880)
  at com.ibm.msg.client.wmq.internal.WMQConnection.<init>(WMQConnection.java:346)
  ... 6 more
Caused by: com.ibm.mq.jmqi.JmqiException: CC=2;RC=2059;AMQ9213: A communications error for
  occurred. [1=java.net.ConnectException[Connection refused],3=myhost.mydomain]
  at  com.ibm.mq.jmqi.remote.internal.RemoteTCPConnection.connnectUsingLocalAddress(RemoteTCP...
  ...
  at com.ibm.mq.jmqi.remote.internal.RemoteFAP.jmqiConnect(RemoteFAP.java:1510)
  ... 7 more
Caused by: java.net.ConnectException: Connection refused
  at java.net.PlainSocketImpl.socketConnect(Native Method)
  ... 11 more
```

Question 6 – z/OS GTF Example



```
------------------------ IPCS Default Values ---------------------------------V3
Command ===>  IP GTF USR(5E9 5EA)  _

  You may change any of the defaults listed below.  The defaults shown before
  any changes are LOCAL.  Change scope to GLOBAL to display global defaults.

  Scope    ==> LOCAL    (LOCAL, GLOBAL, or BOTH)

  If you change the Source default, IPCS will display the current default
  Address Space for the new source and will ignore any data entered in
  the Address Space field.

  Source   ==> DSNAME('LAURANE.GTF.TRACE7')
  Address Space    ==> RBA
  Message Routing ==> NOPRINT TERMINAL
  Message Control ==> CONFIRM VERIFY FLAG(WARNING)
  Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL

Press ENTER to update defaults.

Use the END command to exit without an update.

PF13=MQFMT    14=MQSCAN   15=MQSAVECH 16=MQREGS   17=MQWHERE  18=MQEYEB
PF19=MQINCL   20=MQXPAND  21=MQEXCL   22=MQWHATIS 23=MQTOD    24=MQRUNC
MA     e                                                             02/037
```

Question 6 – z/OS GTF: MQGET Call

- This trace is modified slightly to make the lines fit

Call

```
USRD9 5E9 ASCB 00FADA00            JOBN LAURANEG
  CSQW072I ENTRY: MQ user parameter trace
  GETMSG
    Thread... 00000000  Userid... 00000000  00000000            Hobj..... 00000001
    pMsgDesc. 13B596B0  pGMO..... 13B59518
    BufferL.. 00000032  pBuffer.. 13B79058  DataL.... 00000000  pECB..... 13B89CEC
    RSV1..... 00000000  RSV2..... 00000000  RSV3..... 00000000  QRPLX.... 00000000
                GMT-03/18/2010 03:40:21.575271  LOC-03/18/2010 03:40:21.575271
```

Message Descriptor

```
USRD9 5E9 ASCB 00FADA00            JOBN LAURANEG
  CSQW072I ENTRY: MQ user parameter trace
   +0000  D4C44040  00000002  00000000  00000008  | MD  ............ |
   +0010  FFFFFFFF  00000000  00000311  00000000  | ................ |
   +0020  40404040  40404040  FFFFFFFF  00000002  |         ........ |
   +0030  00000000  00000000  00000000  00000000  | ................ |
   +0040  00000000  00000000  00000000  00000000  | ................ |
   +0050  00000000  00000000  00000000  00000000  | ................ |
                GMT-03/18/2010 03:40:21.575274  LOC-03/18/2010 03:40:21.575274
```

Empty MsgId & CorrelId

Get Message Options

```
SRD9 5E9 ASCB 00FADA00            JOBN LAURANEG
  CSQW072I ENTRY: MQ user parameter trace
   +0000  C7D4D640  00000001  00006021  00000000  | GMO ......-..... |
   +0010  FF000000  00000000  40404040  40404040  | ........         |
```

Question 6 – z/OS GTF: MQGET Return

Return

Message Length

```
USRD9 5EA ASCB 00FADA00              JOBN LAURANEG
  CSQW073I EXIT:  MQ user parameter trace
  GETMSG
    Thread... 00000000  Userid... 00000000  00000000        Hobj..... 00000001
    pMsgDesc. 13B596B0  pGMO..... 13B59518
    BufferL.. 00000032  pBuffer.. 13B79058  DataL.... 00000032  pECB..... 13B89CEC
    RSV1..... 00000000  RSV2..... 00000000  RSV3..... 00000000
    CompCode. 00000000  Reason... 00000000
```

Updated Message Descriptor

Returned MsgId & CorrelId

```
USRD9 5EA ASCB 00FADA00              JOBN LAURANEG
  CSQW073I EXIT:  MQ user parameter trace
   +0000  D4C44040  00000002  00000000  00000008  | MD  ............ |
   +0010  FFFFFFFF  00000000  00000311  000001F4  | ................ |
   +0020  D4D8E2E3  D9404040  00000000  00000001  | MQSTR   ........ |
   +0030  D4D8E3D7  E4E361F2  F0F1F0F0  F3F0F9F2  | MQTPUT/201003092 |
   +0040  F3F0F3F5  F4F6F9F1  D4D8E3D7  E4E361F2  | 30354691MQTPUT/2 |
   +0050  F0F1F0F0  F3F0F9F2  F3F0F3F5  F4F6F9F1  | 0100309230354691 |
   +0060  00000000  40404040  40404040  40404040  |                  |
   +0070  40404040  40404040  40404040  40404040  |                  |
   +0080  40404040  40404040  40404040  40404040  |                  |
   +0090  40404040  D9E3D7C8  40404040  40404040  |     RTPH         |
   +00A0  40404040  40404040  40404040  40404040  |                  |
   +00B0  40404040  40404040  40404040  40404040  |                  |
```

Updated Get Message Options

```
  CSQW073I EXIT:  MQ user parameter trace
   +0000  C7D4D640  00000001  00006021  00000000  | GMO ......-..... |
   +0010  FF000000  00000000  D3C1E4D9  C14BD8D3  | ........LAURA.QL |
```

Message Contents

```
USRD9 5EA ASCB 00FADA00              JOBN LAURANEG
  CSQW073I EXIT:  MQ user parameter trace
   +0000  E3C8C9E2  40C9E240  40C140E3  C5E2E340  | THIS IS A  TEST  |
   +0010  D4C5E2E2  C1C7C540  40404040  40404040  | MESSAGE          |
```

Open Lines for Questions

We Want to Hear From You!

Tell us about what you want to learn

Suggestions for future topics
Improvements and comments about our webcasts
We want to hear everything you have to say!

Please send your suggestions and comments to:
wsehelp@us.ibm.com

Summary

- We covered a number of important trace topics across distributed systems, z/OS and also Java and JMS
  - Steps for collecting trace
  - Common issues collecting trace
  - Performance effects of trace
  - Tools for examining trace
  - Explanation of the trace fields
  - Debugging applications with trace

- We hope this presentation will help you to understand WebSphere MQ trace and use it effectively

References and Useful Links

- WebSphere MQ Information Center – strmqtrc

  http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.amqzag.doc/fa16100_.htm

- WebSphere MQ Information Center – Java trace options

  http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm10770_.htm

- SupportPac MP16 – WebSphere MQ for z/OS Capacity Planning and Tuning

  http://www.ibm.com/support/docview.wss?rs=171&uid=swg24007421

- WebSphere MQ MustGather – Directions to start, end and format trace

  http://www.ibm.com/support/docview.wss?rs=171&uid=swg21174924

- WebSphere MQ Support Portal

  http://www.ibm.com/support/entry/portal

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations
  http://www.ibm.com/software/websphere/support/supp_tech.html

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts, and podcasts
  http://www.ibm.com/developerworks/websphere/community/

- Join the Global WebSphere User Group Community
  http://www.websphere.org/

- Access key product show-me demos and tutorials by visiting the IBM Education Assistant
  http://www.ibm.com/software/info/education/assistant

- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically
  http://www.ibm.com/software/websphere/support/d2w.html

- Sign up to receive weekly technical My Notifications emails
  http://www.ibm.com/software/support/einfo.html