Top Ten Problems on WebSphere MQ - Distributed

Greg Bowman (bowmanga@us.ibm.com)
Jack White (jmwhite2@us.ibm.com)
Advisory Software Engineers
28 July 2010
Agenda

- Introduction
- Top Ten
- Summary
Introduction

- What makes up our “Top Ten”
  - Problems that have been reported often
  - Problems that we think may effect more people
  - Problems that just seemed interesting
Introduction

- How we are presenting these items
  - Only giving high level details in most cases
  - “Top Ten” are not in any particular order
  - Specific problems and general topics
  - Based on items from the IBM® WebSphere MQ Support site
1. AMQ9507 when trying to delete a WMQ channel

**Problem:** You delete a WebSphere MQ channel and get error message: AMQ9507: CHANNEL IN-DOUBT. You try the following actions and you continue to get the error.

- Resolve the channel using backout option.
- Resolve the channel using commit option.
- Remove the channel entry in SYSTEM.CHANNEL.SYNCQ

**Cause:** The transmission queue (which had data on it) was deleted, before the channel was deleted. Other conditions may also cause this problem.
1. AMQ9507 when trying to delete a WMQ channel

- **Resolution:**

  1. Make a backup of all your channel definitions. You can use SupportPac MS03 if you do not already have a utility to do this.

  2. Make backup copies of the following files, then delete these files. The paths start with /var/mqm/qmgrs/<qmgrname>/...

     **Version 5.3** : ...
     @ipcc/AMQRFCD.A.DAT
     ./ipcc/AMQRSYN.A.DAT
     ../scratch (this entire directory)

     **Version 6.0** : ...
     @ipcc/AMQRSYN.A.DAT
     ../scratch (this entire directory)

     **Version 7.0** : ...
     @ipcc/AMQRSYN.A.DAT
     ../channel (leave the SYSTEM* entries in this directory)
1. AMQ9507 when trying to delete a WMQ channel

Resolution (continued):
3. Clear the SYSTEM.CHANNEL.SYNCQ.
   
   runmqsc qmgrname
   clear ql (SYSTEM.CHANNEL.SYNCQ)

4. Redefine all your channels using the output from Step 1 above.

Note: *There is no guarantee that his will always work.* If it still fails, it will not be possible to use the channel name again, unless you recreate your queue manager.

- See technote 1194218 for additional details.
2. WMQ Cluster users may see various problems

- **Problem:** WMQ cluster users may encounter a variety of issues if they are not up to date on WMQ maintenance
- **Cause:** There are a number of cluster related apars in the recent MQ fixpacks
  - **IZ59905:** WEBSPHERE MQ V6 32-BIT CLUSTER QUEUE MANAGER MIGRATED TO V7 64-BIT CLUSTER QUEUE MANAGER FAILS TO START 7012
  - **IZ47841:** 2030(MQRC_MSG_TOO_BIG_FOR_Q) ISSUED WHEN PUTTING A MESSAGE TO A REMOTE CLUSTER QUEUE CAPABLE OF ACCOMODATING THE MESSAGE 6029/7012
  - **IZ52974:** POSSIBLE EXCEPTION IN KPIDEBUG WHEN AMQLDMPA RUN WITH -C K OPTION WHEN CLUSTERING IS IN USE. 6029/7010
  - **IZ56211:** PUT STATISTICS NOT GENERATED FOR SYSTEM.CLUSTER.TRANSMIT.QUEUE 6029/7011
2. WMQ Cluster users may see various problems

- **IZ56549**: MESSAGES ARE INCORRECTLY PUT TO SYSTEM.CLUSTER.TRANSMIT.QUEUE WHERE THEY REMAIN FOREVER 6029/7011
- **IZ60606**: PARTIAL REPOSITORY QUEUE MANAGERS REPORT AMQ9511, AMQ9488 AMQ9409.REFRESH CLUSTER FAILS WITH MQRC_OBJECT_CHANGED (2041) 6029/7011
- **IZ65003**: MQRC_INCONSISTENT_PERSISTENCE (2185) WHEN PUTTING GROUPED MESSAGES TO A CLUSTER 6029/7012

**Resolution**: Upgrade to a current WMQ maintenance level
- 6.0.2.9 or 7.0.1.2
3. MQJMS2013: invalid security authentication supplied for WMQ Queue Manager

- **Problem:** You have a servlet bean running in WebSphere Application Server, which calls a JMS connectionFactory to create a QueueConnection in WebSphere MQ. This worked until you upgraded WSAS v6.1.0.27 to v6.1.0.29.

  Now you get javax.jms.JMSSecurityException:
MQJMS2013: invalid security authentication supplied for MQQueueManager at com.ibm.mq.jms.MQConnection.createQM
(MQConnection.java:2532) ...

  The MQJMS2013 error indicates that the userid passed from WebSphere Application Server to WMQ cannot be authenticated.
3. MQJMS2013: invalid security authentication supplied for WMQ Queue Manager

- **Problem (continued):**
  In WSAS the userid that is passed in to WMQ will be based your JMS connection factory and WSAS res-auth property.
  - If WSAS res-auth property is set to Container, the userid comes from a container-managed authentication alias in your JMS connection factory
  - If res-auth is set to Application (default), the userid comes from a component-managed authentication alias
  - If res-auth defaults to Application but no component-managed alias is defined in your JMS connection factory, no authentication alias will be used
3. MQJMS2013: invalid security authentication supplied for WMQ Queue Manager

- **Cause:** A change was introduced in WMQ v6.0.2.5 client (WMQ APAR IZ17062) where the userid that is used to start the application server process is passed to WMQ if no other userid is specified. If this userid is passed to WMQ but this userid is not authorized in WMQ you get the MQJMS2013 error.
  
  - The change to the WMQ client that causes the userid that is used to start the application server to be passed to WMQ is only present in WMQ v6.0.2.5 through v6.0.2.7. APAR IZ49302 in WMQ v6.0.2.8 reverses the change so no userid is passed.
  
  - In WSAS v6.1.0.27 the WMQ client JAR files are at the WMQ v6.0.2.4 level.
  
  - In WSAS v6.1.0.31 the WMQ client JAR files are at the WMQ v6.0.2.8 level.
3. MQJMS2013: invalid security authentication supplied for WMQ Queue Manager

- **Resolution:** To resolve the problem, you can do one of three things:
  1. Set a component-managed authentication alias in the connection factory.
  2. Set the value of the res-auth property to Container, which will ensure that the container-managed authentication alias will be used.
  3. Install a different version of the WMQ client on the same machine as WebSphere Application Server.
    - WMQ client can either be v6.0.2.4 or below, or v6.0.2.8 or higher.
    - If you install a WMQ client, you will need to update the MQ_INSTALL_ROOT variable in your WSAS configuration to point to the directory where the client is installed instead of the directory where the WMQ client JAR files that ship with WSAS are installed.
    - See technote 1423890 for additional details.
4. Intermittent poor performance when MDB gets messages from WMQ queues

- **Problem**: You have MDB's driven by messages from WebSphere MQ queues. Intermittently the MDB's experience a delay where it may take several seconds for the retrieval of each message and then the MDB's resume normal processing. The delayed messages have a higher priority than the messages being browsed.

- **Cause**: The MDB's are doing a browse of the messages prior to doing a MQGET using the MQGMO_BROWSE_NEXT option and it has browsed to a later point in the queue than the point where the high priority messages were committed. Messages committed to the queue before the browse cursor's current position are not gettable by that thread until it issues a new MQGET call using the MQGMO_BROWSE_FIRST option. The delay is due to the time it took to browse the remainder of the queue.
4. Intermittent poor performance when MDB gets messages from WMQ queues

- **Resolving the problem:** There are 3 options to help address this situation
  - 1. Use MSGDLVSQ(FIFO) on the WMQ queue
    - messages delivered in arrival sequence instead of priority
  - 2. Use non-ASF mode on the application server
    - application server simply issues MQGET calls. See developerWorks article "When to use ASF and non-ASF modes to process messages in WebSphere Application Server"
  - 3. Use the custom property eoqtimeout on the application server
    - eoqtimeout is length of time queue agent waits between scans of the queue. See technote titled "Messages remain on JMS destinations for a long time before they are delivered to message-driven beans (MDBs)".
5. Delay during MQDISC call in WMQ v7 clients

- **Problem:** When a WebSphere MQ v7.0 client connects to a v7.0 Queue Manager, a delay of 20 + seconds is observed during the MQDISC call.
  - A trace of the WMQ client shows:
    - `getaddrinfo(): AF_INET & AI_NUMERICHOST: rc=11001 errno=0`
    - 11001 is a TCP/IP error code meaning WSAHOST_NOT_FOUND

- **Cause:** Due to the changes in the v7 client (sharecnv), when the client disconnects, the socket is closed and error information is set up in case we want to issue an error message. The act of setting up the error message information collects the host name / ip address, and does reverse DNS lookup, which takes 12 seconds to fail at each attempt (and 2 different lookups are performed) hence the delay.
5. Delay during MQDISC call in WMQ v7 clients

- **Resolution:** This problem will be resolved in apar IZ78516, but there are 2 work arounds:
  - Set sharecnv 0 on the svrconn channel
  - Add a parameter DNSLookupOnError=No to the TCP stanza of qm.ini.
  - See apar IY64349 for additional details on this parameter.
6. WMQ v7.0.1 queue manager fails to start

**Problem:**
WMQ v7.0.1 queue manager fails to start and the following FDC is generated:

```
Probe Id     : - ZS142006
Component    : - zslEnqueue
Program Name : - strmqm
Comment1     : - /var/mqm/sockets/XYZQM/qmgrlocl/
               XXXXX/Enqueue.lck
```

MQM Function Stack
- zslEnqueue
- xcsFFST

+-------------------------------------------------------------------------+

WebSphere® Support Technical Exchange
6. WMQ v7.0.1 queue manager fails to start

Cause:
There was a design change to the Enqueue locking functionality in WMQ v7.0.1.

Local Fix:
Run amqiclen -xvh before running strmqm command

This will be corrected by APAR IZ78438.
- Once the APAR is closed, it will be HIPER.
- url not available at this time as this is a new APAR
7. Java/JMS application hang when connecting to a queue manager

Java™ thread core dumps will show one or more threads stuck in the methods similar to the following:

"Thread-382" (TID:0x39650A00, sys_thread_t:0x395BEFD0, state:CW, native ID:0x00AD10F5) prio=5
at java/lang/Object.wait (Native Method)
at java/lang/Object.wait(Object.java:199)
at com/ibm/mq/jmqqi/remote/internal/RemoteHconn.receiveAsyncTsh
 (RemoteHconn.java:1912)
at com/ibm/mq/jmqqi/remote/internal/RemoteHconn.receiveTSH
 (RemoteHconn.java:1176)
at com/ibm/mq/jmqqi/remote/internal/system/RemoteConnection .addHconn(RemoteConnection.java:891)
at com/ibm/mq/jmqqi/remote/internal/system/RemoteConnectionPool .getConnection(RemoteConnectionPool.java:495(Compiled Code))
7. Java/JMS application hang when connecting to a queue manager

Java thread core dumps continued:

"Thread-232" (TID:0x3562DB00, sys_thread_t:0x356797E8, state:CW, native ID:0x007DC0C7) prio=5
at java/lang/Object.wait(Native Method)
at java/lang/Object.wait(Object.java:199(Compiled Code))
at com/ibm/mq/jmqi/remote/internal/system/RemoteConnection
    .sendTSH(RemoteConnection.java:2185(Compiled Code))
at com/ibm/mq/jmqi/remote/internal/system/RemoteConnection
    .addHconn(RemoteConnection.java:876(Compiled Code))
at com/ibm/mq/jmqi/remote/internal/system/RemoteConnectionPool
    .getConnection(RemoteConnectionPool.java:490(Compiled Code))
7. Java/JMS application hang when connecting to a queue manager

You may also see the following exception:

```java
java.util.ConcurrentModificationException
    at java.util.LinkedList$List$Itr.checkForComodification(LinkedList.java:617)
    at java.util.LinkedList$List$Itr.next(LinkedList.java:552)
    at com.ibm.mq.jmqi.remote.internal.system.RemoteConnectionPool.getConnection(RemoteConnectionPool.java:173)
```
7. Java/JMS application hang when connecting to a queue manager

**Workaround:**
Disable connection sharing in the WMQ Java/JMS client by:
- using the connection factory .setShareConvAllowed(0)
or
- setting the MQ classes for Java property MQC.SHARING_CONVERSATIONS_PROPERTY to 1

**Fix:** APAR IZ65557

This APAR is included in FixPack 7.0.1.2.
8. WMQ MQ File Transfer Edition users may experience problems

**APAR's included in WMQ FTE FixPack 7.0.2.1**

**IC66887**: WEBSPHERE MQ FILE TRANSFER EDITION AGENT ENDS ABNORMALLY DURING A FAILED MQGET CALL

**IC66063**: WMQ FTE 702 HAS A TIMING ISSUE THAT OCCURS WHEN NEGOTIATION BETWEEN 2 AGENTS ARE TAKING LONGER THAN NORMAL TO COMPLETE

**IC69123**: FTE AGENT IS UNABLE TO START AFTER APPLYING IFIXES FOR APARS IC67508 AND IC68561
8. WMQ MQ File Transfer Edition users may experience problems

**ifixes available:**

**IC67508:** WMQ FTE 7.0.2.1 AGENT(COMMADHANDLER) FAILS AFTER UPGRADE WITH PROBE FFDC_036 AND METHOD PROCESSQUEUEDREQUESTS

**IC68561:** WMQFTE V7.0.2 TRIGGERING CONDITION NOT MET GENERATES FTE REPLY MESSAGES THAT FILL THE DEAD LETTER QUEUE

**IC69465:** WMQFTE V7 GETS BFGDB0003E(ORA-01722) IF PREACTION OR POSTACTION IS USED IN FTEANT TRANSFER
8. WMQ MQ File Transfer Edition users may experience problems

ifixes available continued:

IC68925: WMQFTE V7 PROTOCOL BRIDGE AGENT STARTS TO LOG BFGIO0134E FOR EVERY SCHEDULED TRANSFER AND TRANSFERS DO NOT GET SENT

IC69248: WMQ FILE TRANSFER EDITION 7.0 USING PROTOCOL BRIDGE FAILS TO DELETE SOURCE FILE. SERVER DELETE FAILS WITH 'FILE NOT FOUND'

IC69702: WMQFTE V7 DATABASE LOGGER IS TERMINATED WHEN STARTING WITHOUT -F PARAMETER
8. WMQ MQ File Transfer Edition users may experience problems

**APAR's recently created**

IC69682: WMQ FTE V7.0.2.1 GETS BFGIO0060E WHEN TRANSFERRING A TEXT FILE EVEN THOUGH THE SOURCE AND DESTINATION CODE PAGES ARE THE SAME

IC69796: WMQFTE V7 RESOURCE MONITOR TRIGGERS FOR A LOCKED FILE EVEN THOUGH ENABLEMANDATORYLOCKING PROPERTY IS SET TO TRUE.

IC68886: WMQFTE 7.0.2.X SOURCE AGENT CRASHED WHILE TRANSFERRING A FILE TO THE DESTINATION. AN FFDC WAS GENERATED WITH PROBE ID FFDC_001

IC69713: WMQFTE V7 AGENT TERMINATES UNEXPECTEDLY OR HANGS WITH A BFGDM0082E ERROR WITH JMQI REASON CODE 2397.

IC69703: WMQ EXPLORER HANGS WITH PROBE ID JU:40010, WHILE TRACING IS ENABLED AND THE WMQFTE TRANSFER LOG IS LOST
9. WMQ v7 queue manager hangs when using the broker

Problem:
When using the broker contained within WMQ v7, it is possible for the queue manager to enter a hung state.

Cause:
The hang can occur if a query of the queue managers connections is made either by using the runmqsc interface or by use of the PCF facility while publishing is occurring.
9. WMQ v7 queue manager hangs when using the broker

Example of the sequence of events leading to the hang:

1. A message is published by process 'A' and the topic tree lock is taken.

2. A DIS CONN command is issued by process 'B'. This calls in to the queue manager and an HObj lock is taken.

3. The code under DIS CONN tries to take the topic tree lock so that it can get at the topic string associated with topic object handles opened for publishing. This lock is owned by process 'A' and so it waits.

4. Process 'A' tries to take the HObj lock but this is held by process 'B' so it waits.

5. A deadlock is now in effect which causes the queue manager to hang.
9. WMQ v7 queue manager hangs when using the broker

Local Fix:
Avoid using DISPLAY CONN or DISPLAY QSTATUS TYPE(HANDLE)

This will be fixed by IZ79335
- url not available at this time as this is a new APAR
10. WMQ v7 client using conversation sharing fails with a memory access violation

**Problem:**
A client application using conversation sharing fails during an MQ connect call due to an error when starting a thread. The application then ends abruptly with a memory access violation.

**Symptoms:**
- AMQ9233: Error creating receive thread.

**EXPLANATION:**
The process attempted to create a new thread. The most likely cause of this problem is a shortage of an operating system resource (for example: memory). Use any previous FFSTs to determine the reason for the failure.
10. WMQ v7 client using conversation sharing fails with a memory access violation

**Symptoms on Windows®:**

- First FDC will show:
  
  Probe Id :- XC035007  
  Component :- xcsCreateThread

- Second FDC will show:
  
  Probe Id :- XC130031  
  Component :- xehExceptionHandler
10. WMQ v7 client using conversation sharing fails with a memory access violation

**Symptoms on Unix®**

- **First FDC**
  - AIX, HP-UX and Solaris
    - Probe Id: : X035011
    - Component: - xcsCreateThread
  - Linux®
    - Probe Id: : X035039 (possibly X035040)
    - Component: - xcsCreateThread

- **Second FDC**
  - Probe Id: : X130003
  - Component: - xehExceptionHandler
10. WMQ v7 client using conversation sharing fails with a memory access violation

**Symptoms continued:**

- Function stack in first FDC for both Windows and Unix will end with:
  - ccxSetAsyncMode
  - xcsCreateThread
  - xcsFFST

- Function stack in second FDC for both Windows and Unix will end with:
  - ccxFreeSocket
  - xcsFFST
10. WMQ v7 client using conversation sharing fails with a memory access violation

Workaround:
Disable conversation sharing.
Example: set SHARECNV(0) on the SVRCONN channel

This will be fixed by APAR IC66174 in FixPack 7.0.1.3.
11. WebSphere MQ v6

- End of Service date: 30 September 2011

  See technote 1418724

- This includes v6.0.1 and v6.0.2 on the distributed platforms
11. WebSphere MQ v6

References:

- Migrating to WebSphere MQ V7
- Installation and Migration of WebSphere MQ V7 including Publish/Subscribe Configuration
Summary

- Keep current with FixPacks
- Check the WMQ Support Portal
Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:

- Join the Global WebSphere Community:
  http://www.websphereusergroup.org

- Access key product show-me demos and tutorials by visiting 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:
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
Questions and Answers