

# Unit 18. Using traces, error logs, and failure data capture files

## What this unit is about

This unit covers the WebSphere MQ trace mechanism, explains the contents of the AMQERR01.LOG file, and discusses the first failure support technology. Problem determination hints and tips are also provided for some of the more common types of problems.

## What you should be able to do

After completing this unit, you should be able to:

- Explain the meaning of entries in the AMQERR01.LOG file
- Identify where failure data capture (FDC) files can be found
- Explain how to use WebSphere MQ trace to solve development problems
- Describe some of the more common problem types and how to approach initial problem determination

## How you will check your progress

- Lab exercise

## References

- Intercommunication Guide
- WebSphere MQ Application Programming Reference
- Quick Beginnings Guide
- WebSphere MQ Using Java
- WebSphere MQ System Administration Guide

## Unit objectives

After completing this unit, you should be able to:

- Explain the meaning of entries in the AMQERR01.LOG file
- Identify where failure data capture (FDC) files can be found
- Explain how to use WebSphere MQ trace to solve development problems
- Describe some of the more common problem types and how to approach initial problem determination

© Copyright IBM Corporation 2008

Figure 18-1. Unit objectives

WM203 / VM2032.0

### Notes:

## Configuration files and problem determination

- Errors can stop a queue manager being found
  - QUEUE MANAGER UNAVAILABLE
- What to check
  - Configuration files exist
  - Configuration files have the appropriate permissions
  - WebSphere MQ configuration file (or Windows registry) references the queue manager with the correct information for locating its files

© Copyright IBM Corporation 2008

Figure 18-2. Configuration files and problem determination

WM203 / VM2032.0

### Notes:

If you receive an error message indicating that the queue manager is unavailable, the cause could be something simple like the queue manager has not been started or an application specified an incorrect queue manager name. If the cause is not anything obvious, check the items listed on the visual. Errors in a configuration file can typically prevent a queue manager from being found.

- Ensure that the configuration files exist.
- Ensure that they have the appropriate permissions. For example, on a UNIX system, the WebSphere MQ configuration file should have the following permissions.  
`-rwxrwxr-x 1 mqm mqm 1371 Sep 17 14:32 /var/mqm/mqs.ini`
- Ensure that the WebSphere MQ configuration file references the queue manager and has the correct information for locating the files associated with it.

## Error log

- Captures messages concerning:
  - Operation of WebSphere MQ
  - Any queue managers that you start
  - Error data coming from the channels that are in use
- Location depends on whether queue manager name is known and whether error is associated with a client
  - If queue manager name is known and queue manager is available:
    - Windows: mqmtop\QMGRS\QMgrName\ERRORS\AMQERR01.LOG
    - UNIX: /var/mqm/qmgrs/QMgrName/errors/AMQERR01.LOG
  - If an error has occurred with a client application:
    - Windows: mqmtop\ERRORS\AMQERR01.LOG
    - UNIX: /var/mqm/errors/AMQERR01.LOG
- Errors subdirectory can contain up to three error log files
  - AMQERR01.LOG, AMQERR02.LOG, AMQERR03.LOG

© Copyright IBM Corporation 2008

Figure 18-3. Error log

WM203 / VM2032.0

### Notes:

Error messages are only written to an error log file called AMQERR01.LOG. There is a separate error log file with this name in each of three error directories as described in the figure.

When the error log file AMQERR01.LOG fills up, its contents are copied to AMQERR02.LOG and AMQERR01.LOG is then reused. Before the copy, AMQERR02.LOG is copied to AMQERR03.LOG. The previous contents of AMQERR03.LOG, if any, are discarded. In this way, AMQERR02.LOG and AMQERR03.LOG are used to maintain a history of error messages.

On Windows, also examine the Windows application event log for relevant messages.

In the figure, in the directory location for Windows, replace mqmtop with the appropriate directory, probably C:\Program Files\IBM\WebSphere MQ\.

## First failure support technology (FFST)

- “Unexpected” errors
  - Internal queue manager failure but not automatically an APAR
  - May have associated storage dumps
  - Data to help analyze software events
- If an error occurs:
  - Note a description of the problem
  - Look for any related error log entries
  - Identify any FFST reports
- Location of FDC files:
  - On UNIX platforms, look in directory:  

/var/mqm/errors
  - Windows:  

c:\Program Files\IBM\WebSphere MQ\errors

© Copyright IBM Corporation 2008

Figure 18-4. First failure support technology (FFST)

WM203 / VM2032.0

### Notes:

- FFST is used to record an internal problem at the point at which it is discovered.
- WebSphere MQ uses FFST as follows:
  - To detect and report software events, for example, internal queue manager failures.
  - To collect information about software events, for example, dumps of storage.
  - To generate data to help analyze software events, for example, probe IDs.
- Each FFST report contains various items of useful information.
  - A probe ID which identifies where in the code the error was detected.
  - The date and time the error occurred.
  - Any associated error message.
  - A variable number of dumps including the function stack and trace history.
- If you experience frequent FFST reports related to shared memory on a UNIX system, it may mean that you need to change the values of certain kernel parameters in order to

support WebSphere MQ. For more details on which kernel parameters may need changing, refer to *WebSphere MQ for HP-UX, Sun Solaris, or AIX Quick Beginnings*, or the relevant System Management Guide.

- FDC files are named AMQnnnnnnnn\_mm.FDC, where: *nnnnnnnn* is the process ID reporting the error *mm* is a sequence number, normally 0.

## FFST report example

```
+-----+
| WebSphere MQ First Failure Symptom Report
+-----+
| Date/Time      :- Mon January 28 2008 21:59:06 GMT
| UTC Time/Zone :- 1201539869.892015 0 GMT
| Host Name     :- 99VXY09 (Windows XP Build 2600: Service Pack 1)
| PIDS          :- 5724H7200
| LVLS          :- 7.0.0.0
| Product Long Name :- WebSphere MQ for Windows
| Vendor        :- IBM
→ | Probe Id      :- HL010004
→ | Application Name :- MQM
→ | Component    :- hlgReserveLogSpace
→ | SCCS Info   :- lib/logger/amqhlge0.c, 1.26
→ | Line Number  :- 246
→ | Build Date   :- Jan 25 2008
→ | CMVC level   :- p000-L050202
→ | Build Type   :- IKAP - (Production)
→ | UserID        :- IBM User
→ | Process Name :- C:\Program Files\IBM\WebSphere MQ\bin\amqzlaa0.exe
→ | Process       :- 00003456
→ | Thread        :- 00000030
→ | QueueManager :- qmgr2
→ | ConnId(1) IPCC :- 162
→ | ConnId(2) QM  :- 45
→ | Major Errorcode :- hrcE_LOG_FULL
→ | Minor Errorcode :- OK
→ | Probe Type    :- MSGAMQ6709
→ | Probe Severity:- 2
→ | Probe Description :- AMQ6709: The log for the Queue manager is full.
→ | FDCSequenceNumber :- 0
+-----+
```

© Copyright IBM Corporation 2008

Figure 18-5. FFST report example

WM203 / VM2032.0

### Notes:

This slide shows an FFST or FDC file which was cut due to a log file full condition on a Windows WebSphere MQ V7 system. Some of these fields which are useful are:

- Probe Id - A unique error code and identifies where in the code the error was detected
- Component - The function that failed
- Process name – The name of the process executing at the time of the failure
- Major error code – The named reason for the failure
- Probe Description – The externalized message id and meaning which is written to the error log

## Using trace

---

- Extra information may be needed to find a problem
  - Files can be very large
  - Can be limited by time or component
- Can also trace the MQI
  - Useful aid to application debugging
- Trace commands:
  - Start trace **strmqtrc**
  - Stop trace **endmqtrc**
  - Format trace **dspmqtrc**

© Copyright IBM Corporation 2008

Figure 18-6. Using trace

WM203 / VM2032.0

### Notes:

The command to start a trace is **strmqtrc**

The command to end a trace is **endmqtrc**

## Trace commands

- To start, stop and format trace:
  - Trace files are written to the directory /var/mqm/trace on UNIX
  - For Windows, C:\Program Files\IBM\WebSphere MQ\trace
  - delete or relocate old trace files before beginning a new trace.
  
- Start trace
  - For every WebSphere MQ process:  
 > strmqtrc -e
  - Or start trace only for one queue manager:  
 > strmqtrc -m MY.QMGR
  - Or start a high detail trace for one queue manager:  
 > strmqtrc -t all -t detail -m MY.QMGR
  - Or start a high detail wrapping trace and limit the file size to ~5MB:  
 > strmqtrc -l 5 -t all -t detail -m MY.QMGR
  
- End all tracing:  
   > endmqtrc -a
  
- Format the trace files on UNIX:
  - Or format wrapping trace files:  
 > dspmqtrc \*.TRC
  - Or format wrapping trace files:  
 > dspmqtrc \*.TRC \*.TRS

© Copyright IBM Corporation 2008

Figure 18-7. Trace commands

WM203 / VM2032.0

### Notes:



#### On UNIX platforms:

- From WebSphere MQ v6.0, AIX now supports the standard WebSphere MQ trace facility as well as the AIX system trace. It is recommended that V6.0 customers use the WebSphere MQ trace commands unless requested by the WebSphere MQ service team to use the AIX system trace hooks.
- The trace formatter program converts binary files named AMQppppp.TRC (where ppppp is the process identifier or pid which created the file) into readable files named AMQppppp.FMT.
- If you used a wrapping trace, then each time a .TRC reaches the size limit WebSphere MQ renames it to a .TRS extension and starts a new .TRC file. The trace formatter can

convert both files to a single formatted file, but only if you format the .TRC and .TRS files at the same time, as shown in the slide.



### **Windows**

The default trace directory is C:\Program Files\IBM\WebSphere MQ\trace.

On Windows, the trace files are readable without formatting. However, they are still called AMQppppp.TRC (where ppppp is the process identifier or pid which created the file) rather than AMQppppp.FMT.

## Example trace on WebSphere MQ for Solaris

```

Timestamp Process.Thread Trace Data
=====
15:00:04.324190 12277.1      Version : 6.0.0.0 Level : p000-L050203
15:00:04.325045 12277.1      Date   : 07/02/05 Time  : 15:00:04
15:00:04.325375 12277.1      PID    : 12277 Process : strmqm
15:00:04.325403 12277.1      QueueManager : QM1
15:00:04.325419 12277.1
15:00:04.325446 12277.1      Trace Control Memory:
15:00:04.325471 12277.1      StrucId:
15:00:04.325490 12277.1      EarlyTraceOptions: 0
15:00:04.325507 12277.1      EarlyTraceMaxFileSize: 0
15:00:04.325527 12277.1      ActiveEntries: 0
15:00:04.325544 12277.1      Options MaxFileSize FileCount SubPoolName
15:00:04.325566 12277.1      74fffff          0 0 elk
15:00:04.325587 12277.1      0           0 0
15:00:04.325609 12277.1      0           0 0
15:00:04.325632 12277.1      0           0 0
15:00:04.325654 12277.1      0           0 0
15:00:04.325677 12277.1      0           0 0
15:00:04.325698 12277.1      0           0 0
15:00:04.325774 12277.1      0           0 0
15:00:04.325798 12277.1      0           0 0
15:00:04.325891 12277.1      Thread stack
15:00:04.325971 12277.1      -> zslWaitEC
15:00:04.326078 12277.1      -> zslCheckIfRunning
15:00:04.326098 12277.1      -> xcsInitialize
15:00:04.326147 12277.1      -> xcsGetEnvironmentString
15:00:04.326186 12277.1      ---{ xcsGetEnvironmentString
15:00:04.326241 12277.1      xcsGetEnvironmentString[AMQ_SERVICE_MODULE] = NULL

```

© Copyright IBM Corporation 2008

Figure 18-8. Example trace on WebSphere MQ for Solaris

WM203 / VM2032.0

### Notes:

Trace output is unformatted on UNIX systems; use the **dspmqtrc** control command to format trace output before viewing. For example, to format all trace files in the current directory use the following command:

**dspmqtrc \*.TRC**

## Tips for channel problems

- Verify the listener program is running on the receiving end of the channel
  - For /etc/services, and inetc.conf, check the configuration
- Verify that the queue manager is running.
- Verify the connection from sender to receiver using the native TCP/IP ping command.
- Channels come in pairs and they must both have the same name, and CASE MATTERS.
- Check the error logs on the sending and receiving side of the channel.
- Try the following MQSC commands on the sending side of the problem channel. These commands can correct several channel related problems.
  - stop channel(*channel\_name*)
  - reset channel(*channel\_name*) seqnum(1)
  - resolve channel(*channel\_name*) action(backout or commit)
  - start channel(*channel\_name*)

© Copyright IBM Corporation 2008

Figure 18-9. Tips for channel problems

WM203 / VM2032.0

### Notes:

Regarding the listener, some environments use /etc/services, and inetc.conf instead of the **runmqlsr** 'listener'. You must verify that the files are configured correctly. Details regarding that configuration are provided in the *Intercommunication Guide*.

## Dead-letter queue

The DLQ holds undelivered messages.

1. Browse the message on the dead-letter queue.
2. Locate the dead letter header.
  - The dead letter header starts with the characters "DLH".
  - Example:  
`444C 4820 0000 0001 0000 0108 4D59 2E51 'DLH .....MY.Q'`
3. Find the dead letter reason
  - Third word in the dead letter header. In this example the dead letter Reason is: x'0000 0108'.
4. Convert hex x'0000 0108' to decimal 264.
5. Use the dead letter reason code to determine why the message was placed on the dead-letter queue.
  - Example: **MQFB\_CHANNEL\_FAIL 264 X'00000108'**

© Copyright IBM Corporation 2008

Figure 18-10. Dead-letter queue

WM203 / VM2032.0

### Notes:

The dead letter reason provides a code to identify the reason the message arrived on the DLQ. It reveals a Reason Code (MQRC\_\*) or a Feedback Code (MQFB\_\*)�

## Java and JMS tips

- Verify correct path, classpath and environment variables
  - The `java.lang.NoClassDefFoundError`: is normally a classpath problem.
- Java client connectivity issues treated like any other channel failure.
- Check queue connection factory (QCF) or topic connection factory (TCF) is configured to connect to the server where your queue manager resides.
- Verify that the queue manager and listener programs are running.
- Verify that the server connection channel is defined on the server to support the queue and topic connection.
- Test the client connection using WebSphere MQ Java client or C client sample programs.
- Try a TCP/IP ping from client to server.

© Copyright IBM Corporation 2008

Figure 18-11. Java and JMS tips

WM203 / VM2032.0

### Notes:

Most WebSphere MQ failures are caused by setup and configuration problems. The path, class path, and environment variables must be configured as per Chapter 2 of the *WebSphere MQ Using Java* manual.

## Remote administration failures

Using MQ Explorer for remote administration:

- Check the Command server is running on remote queue manager.
- If the Listener is not running on remote queue manager, message AMQ4043 will be issued.
- Check **SYSTEM.ADMIN.SVRCONN** is defined on remote queue manager.
- If security check fails, message AMQ4043 will be issued.
- Code page conversion failures can cause problems. Message AMQ4043 will be issued.
- The **SYSTEM.MQEXPLORER.REPLY.MODEL** is not defined on remote queue manager and you are using WebSphere MQ Explorer V6.0.
  - Message AMQ4400 will be issued.

© Copyright IBM Corporation 2008

Figure 18-12. Remote administration failures

WM203 / VM2032.0

### Notes:

Common remote administration failures are:

- **AMQ4043** - Queue manager not available for connection.
- **AMQ4400** - Explorer cannot administer the queue manager because the queue <insert\_0> is not defined.

Ensure that you have satisfied the following requirements before trying to use the WebSphere MQ Explorer to do remote administration. Verify that:

- The WebSphere MQ server and client are installed on the local and the remote computer.
- A command server is running for every queue manager.
- A TCP/IP listener exists for every queue manager. It can be the WebSphere MQ listener or the inetd daemon as appropriate for your operating system environment.

- The server-connection channel, called SYSTEM.ADMIN.SVRCONN, exists on every remote queue manager. This channel is mandatory for every remote queue manager being administered.

Create the channel using the following MQSC command: **DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)**.

- The user ID of the initiator must be a member of the "mqm" group on the local and remote computer. The model queue SYSTEM.MQEXPLORER.REPLY.MODEL must exist on every queue manager.

Create the queue using the following MQSC command: **DEFINE QMODEL(SYSTEM.MQEXPLORER.REPLY.MODEL)**



### Note

A note about Message variable notation:

Some messages display text or numbers that vary according to the circumstances and generating the message; these are known as message variables. The message variables are indicated as <insert\_1>, <insert\_2>, and so on.

In some cases a message might have variables in the Explanation or Response. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the Response, is recorded there.

## Channel triggering problems

- Verify that the channel initiator is running. It is the most important piece of the triggering process.
- Use the **runmqchi** command to run a channel initiator process.
  - Example:

```
runmqchi -q SYSTEM.CHANNEL.INITQ -M CLIENT
```

- Make sure that the channel initiator is monitoring the initiation queue, not the transmission queue.
- Check the error log for channel error messages.
- Try to start the channel manually.
  - If the channel fails to start, or does not successfully move the message from the transmission queue to the remote queue manager, then this is a channel problem. Debug the channel problem.

© Copyright IBM Corporation 2008

Figure 18-13. Channel triggering problems

WM203 / VM2032.0

### Notes:

When debugging a channel triggering problem, it is a good practice to set a short disconnect interval on the associated channel. The disconnect interval setting will stop the channel quickly, with triggering enabled, and make debugging easier.

## Triggering program problems

---

- Make sure that the trigger monitor is running
  - Check it is monitoring the initiation queue, not the trigger queue,
- Verify that your applications are putting their messages to the trigger queue, not the initiation queue.
- Try to start the trigger program manually using the string specified in the applicid property of your process definition.
- Verify that the user ID used to start the trigger monitor has the authority to access the entire path to the executable file.

© Copyright IBM Corporation 2008

Figure 18-14. Triggering program problems

WM203 / VM2032.0

### **Notes:**

## Checkpoint questions

1. What is the name of the file that contains error messages for WebSphere MQ?
2. What is an FDC and where is it found?
3. What are the three commands to start, end and format a trace?
4. What is contained in the dead-letter queue?

© Copyright IBM Corporation 2008

Figure 18-15. Checkpoint questions

WM203 / VM2032.0

### Notes:

## Unit summary

Having completed this unit, you should be able to:

- Explain the meaning of entries in the AMQERR01.LOG file
- Identify where failure data capture (FDC) files can be found
- Explain how to use WebSphere MQ trace to solve development problems
- Describe some of the more common problem types and how to approach initial problem determination

© Copyright IBM Corporation 2008

Figure 18-16. Unit summary

WM203 / VM2032.0

### Notes: