



Making Use of WebSphere MQ Recovery Logs

Author:
Christian Wolfhagen
Senior Development Consultant
Cressida Technology Ltd.

White Paper

Introduction

WebSphere MQ (WMQ) creates log files that allow you to recover queues when there has been a hardware or WMQ failure. Writing these logs is not cheap in terms of system resources, but there are a number of potential benefits to having these logs. Unfortunately WMQ does not allow us to capitalize on most of these benefits.

The information in the recovery logs could be used in a number of areas:

- Statistical information
- Tracking Messages
- Auditing
- Application Analysis and Regression Testing
- Recovery
- Accounting

WebSphere MQ Recovery Log File Content

When you put a persistent message to a queue, WMQ does the following:

- Writes your message to a disk file that represents the queue.
- Writes your message to the log.

Non-persistent messages are not secured to disk.

When you Get and remove a persistent message from the queue, WMQ marks the message in the disk file as being read, and writes another record to the log to indicate that the message has been removed.

As well as Puts and Gets, WMQ logs transaction control, queue manager start and stop events, queue purge events, changes to the configuration of the queue manager or any queues, namelists, processes and topics (MQ Version 7 only) defined under the queue manager, and media images.

What does IBM give you

There are two tools:

- rcrmqobj** which recovers queues and other objects if they have been corrupted by a hardware failure or software failure. To use then features you must have first employed rcdmqimg to record a media image of the object you may be required to recover.
- dmpmqlog** which dumps the last part of the log file to be written. To use the utility the queue manager must be shutdown, a not very likely option in a production environment. The actual output is not very user friendly as can be seen from the sample record below.

White Paper

LOG RECORD - LSN <0:0:3:54623>

HLG Header: lreclsize 718, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
Eyecatcher . . . : ALRH Version : 1
LogRecdLen . . . : 698 LogRecdOwnr . . . : 256 (AQM)
XTranid : TranType: MQI TranNum{High 0, Low 13}
QueueName : MyQueue1
Qid : {Hash 3214482040, Counter: 0}
ThisLSN : <0:0:0:0>
PrevLSN : <0:0:3:53905>

Version : 4
MapIndex . . . : 3
PrevLink.Locn . : 2568 PrevLink.Length : 8
PrevDataLink . . : {High 0, Low 3072}
Data.Locn . . . : 3072 Data.Length . . . : 458

Data :
00000: 41 51 52 48 04 00 00 00 FF FF FF FF FF FF FF FF AQRH.....
00016: 00 00 00 00 00 00 00 00 03 00 00 00 01 00 C0 01 À.
00032: 00 00 00 00 00 00 01 00 0A 00 00 00 00 00 00 00
00048: 05 00 00 00 30 30 30 30 30 30 30 30 30 30 30 30 000000000000
00064: 30 30 30 30 30 30 30 30 30 30 35 37 00 00 00 00 000000000057....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Potential Log File Uses

There are a number of important functions that could be provided only if we could access the WMQ Recovery log data. Using these Recovery logs means that we could access this information in a non-intrusive manner, no Application or infrastructure changes would be needed, assuming persistent messaging was already in use. All of this information becomes available at no extra cost, and remains available until you remove the relevant Recovery log files.

Audit

As every operation on a persistent queue is logged, you can get a list of messages that were written to a queue, and you can see whether or not the messages have been removed from the queue. For Puts you can tell when the message was written and which application put the message on the queue. Unfortunately, there is no way of determining which application removed the message from the queue. But you can tell when the message was removed from the queue, and whether or not it expired.



White Paper

Accounting

If your organization charges per message or per character exchanged, you can use the information in the logs as the basis of charge back. Without using the log information, you have to collect data about messages transferred when they're sent. If you could use the information in the logs, you could run a batch process each night that runs a report about the number of messages and amount of data transferred by target queue, by sending application or any other set of criteria that allows you to identify a specific customer.

If the customer subsequently queries the charging, you can then reproduce the log files as evidence that the charging is correct.

Statistical Information

Because we have a record of every (persistent) message written and removed from every queue, the log file could be used to get lots of statistical, performance and tracing information. The information in the logs can answer questions such as:

- How many messages were put on a queue by each application or location?
- How many messages were written to a queue by hour?
- How many messages were removed from the queue by hour?
- How long were messages on the queue on average?
- What was the longest time that a message remained on a queue?

Tracking Messages

By looking in multiple logs, we could even trace a message across a network and see the response come back again. We could then see how long the message takes at every point in its path through the network, and detect potential bottlenecks. We could also see what happened if an application sent a message but doesn't get a response – we could see where the message got to before it got lost.

You could use this information to:

- Verify claims from customers that your network does not comply to agreed response times.
- Check claims from customers that a message was not delivered.

Application Analysis and Regression Testing

The log files contain every message that was put by an application. The information in the log file is sufficient to recreate the message, which could then be placed on a queue.

Using the log, we could then re-run a complete set of messages through a new version of the application in a test environment before putting it into production.



White Paper

If an application fails to process a message in a production environment, it can be very difficult and expensive to determine what caused the problem. A lot of detective work is need to reproduce the problem – especially if there were many users connected at the time, and any one of them might have caused the problem. The log file keeps a track of the messages that were got from the queue – we could use this to find the actual message that caused the problem and then re-run this message on the test system to reproduce the problem.

Recovery

Recovery works well if you want to recover everything up to the current point in time. The recovery protects you against hardware failures and WMQ failures. It does not protect you against application problems.

If you upgraded an application at 2:00am, and you found 12 hours later that the application was faulty, then all transactions that happened in the last 12 hours should ideally be rolled back.

You can roll back your database to this point in time, but WMQ doesn't provide a facility to roll back queues to a given time. Ideally, you would roll back the queues to the time when the new application was loaded. After that you could potentially re-run of the messages received by the application since it failed.

In general, the logs are an essentially facility to protect against system failures. WMQ does not allow us to make the best possible use of the information in these logs.

Conclusion

There is a wealth of information available in the Recovery logs maintained by WebSphere MQ, if only we could get to it and translate it to a readable and usable format. Cressida Technology has produced its solution, ReQuest™ for WebSphere® MQ, to do just that.

Cressida ReQuest™ for WebSphere® MQ is a new Message Tracking and Auditing, Point-in-Time Recovery, detailed Message Reporting and flexible Message Reply solution utilizing the critical information already captured in the WebSphere MQ Logs.

Cressida ReQuest™ is Generally Available and implemented at several client production installations.

Key Features

- Selective Recovery of Messages and Queues
- Valid Recovery Point Detection, Time Stamped Recovery
- Easy Tracking of Missing or Delayed Messages
- Complete Breakdown of End-To-End Message Response Times
- Auditing, Charge-Back and Accounting Based on Message Content
- Replay Message Activity for System and Load Testing
- No application changes or pre-processing of log file data is required

CDB Software, in Partnership with Cressida Technology Ltd, offers innovative solutions for Websphere MQ log analysis and message monitoring.

About CDB Software

CDB Software, Inc. is a leader in data management solutions for DB2 z/OS. CDB provides unique and innovative solutions that enable companies to expand their systems to meet business needs while controlling the overall cost of the mainframe. Founded in 1985, CDB is a privately held corporation based in Houston, Texas with offices worldwide.

For more information visit:

www.cdbsoftware.com

About Cressida Technology Ltd.

Cressida Technology's solutions focus on on-going research and matching of your requirements with high quality and reasonably priced security and business assurance solutions. Cressida has their Head Office in the United Kingdom.

For more information visit:

www.cressidatechnology.com



CDB Software, Inc.
11200 Richmond Ave.
Houston, TX 77082