**EMPLOYEE DETAILS**

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Phone Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### **DESCRIPTION OF INCIDENT:** Memory Problems

**Location:**

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Policy Notified Yes No

**Incident Details:**

Garbage Collector and memory leaks are to blame for memory issues.

**Incident Causes:**

Garbage collection force all threads to pause so that memory may be reclaimed. There is a concern when this operation takes too long or occurs too frequently. CPU spikes and long reaction times are the most common symptoms.

**Recommendations:**

configuring the -verbosegc parameters, using a performance monitoring tool to detect large GC events, and monitoring heap use and any CPU spikes with a tool. It’s tough to prevent this issue.

**Follow Up Actions:**

mitigate it by adjusting heap size and restarting your JVM.

Assigned To: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reported By:**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Position: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Department: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EMPLOYEE DETAILS**

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Phone Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### **DESCRIPTION OF INCIDENT:** Concurrency Problems

**Location:**

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Policy Notified Yes No

**Incident Details:**

When several calculations are run at the same time, this is known as concurrency.

**Incident Causes:**

Thread deadlocks happen when two or more threads try to access the same resource while waiting for the other to release it, and vice versa. When a deadlock occurs, the JVM exhausts all threads, slowing down the program. It’s pretty tough to recreate a deadlock.

**Recommendations:**

Capturing a thread dump while two threads are blocked and examining [**stack traces**](https://seagence.com/blog/understand-java-stack-traces/) of the threads is one approach to address a deadlock situation.

**Follow Up Actions:**

make the program and its resources as immutable as feasible, utilize synchronization, and check for potential thread interactions.

Assigned To: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reported By:**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Position: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Department: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EMPLOYEE DETAILS**

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Phone Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DESCRIPTION OF INCIDENT:** AMQ7469 Transactions rolled back to release log space

**Location:**

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Policy Notified Yes No

**Incident Details:**

The log space for the queue manager is becoming full. One or more long-running transactions have been rolled back to release log space so that the queue manager can continue to process requests.

**Incident Causes:**

The size of the log depends primarily upon the duration of the longest running Unit Of Work (UOW) and the throughput on the log. When the log starts to become full, then the queue manager will start backing out long running transactions and penalizing (performance) heavy log users.

**Recommendations:**

* You can avoid this problem by committing Units Of Work (UOWs) more frequently.
* Increase the number of log files.

This can be done with the system running and will require a queue manager restart.

**Follow Up Actions:**

Windows:  
Use the MQServices MMC to increase the number of Files.

UNIX and Linux:

* + 1. Edit the /var/mqm/qmgrs/{QMgrName}/qm.ini file. Increase the value of the LogPrimaryFiles & LogSecondaryFiles. The sum of both have to be less than or equal to 63 (V5.3) 511(V6.0)
    2. Stop the QueueManager
    3. Restart QueueManager
* Increase the size of the log files.

This method requires you to delete and redefine the queue manager with larger log file sizes.

Assigned To: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reported By:**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Position: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Department: \_\_\_\_\_\_\_\_\_\_\_\_\_\_