Sample MQSeries Troubleshooting Guide for Request / Reply applications.

By Michael F Murphy
mmurphy@mqsolutions.com

This guide is written for operations personnel with little or no MQSeries experience.
Commands are for Unix or Windows Queue managers but the concepts carry across other platforms

NOTE: This is a recreation from memory of a document previously written. It therefore should be considered a draft and comments are welcome.

This is a sample troubleshooting guide that is shared between two companies using MQSeries to move data. This guide is most useful if you have operations people with little or no MQSeries experience are required to perform first level support. This document is written in a way that if the operations personnel at both companies have a copy, they can both follow along step by step without stepping on each others efforts to resolve the problem. It also helps reduce companies blaming each other for the problem for hours rather than just resolving the problem.

Company A sends requests to Company B and receives replies back from Company B. Company B receives requests from Company A and sends back replies.

If your application is not receiving responses to requests, follow these instructions to troubleshoot and isolate MQSeries problems or eliminate MQSeries as the cause.

MQSC commands are prefixed with MQSC>. To enter interactive mode with MQSeries, make sure you are logged in to a user with mqm privileges and type runmqsc.

The following steps, 1 through 4, are to be performed by Company A.
Step 10 will also be performed by company A. Company A can perform Step 10 now or after Company B has completed its steps.

Checking application logs and dead letter queue.

Step 1:
Check your applications logs to ensure there are not any negative errors being returned by MQSeries. If there are errors, take note of the error. If the error points you directly to the cause of the problem, resolve the problem and check the application again. Otherwise, continue with Step 2.

Step 2:
Check your dead letter queue for messages. If any exist, see if they are coming from your application and check the reason. Resolve the reason. If there are not any messages in the dead letter queue related to this application, proceed to Step 3.

Checking if message are being transmitted to Company B.

Step 3:
Check the depth of the transmit queue that goes to Company B.

```
MQSC> Dis ql(companyb) curdepth
```

If the depth is 0, check the last message date/time sent on your sender channel.

```
MQSC> dis chs(companya.companyb) lstmsgda lstmsgti
```

If the date/time is recent, your messages are being transmitted to Company B. **Skip to Step 5**.

If the date/time is not recent, your messages may be in the dead letter queue. You should have checked the dead letter queue in Step 2.

If the transmit queue depth is 0, your channel does not show a recent send date / time, and your messages are not in the dead letter queue, you most likely have an application problem.

If the depth is greater than 0, you most likely have a problem with the sender channel to Company B. **Proceed with Step 4**.

Step 4:
Check the status of your sender channel to Company B.

```
MQSC> Dis chs(companya.companyb)
```

If the status is STOPPED, try to determine why it is stopped. It can be stopped by a person or by the queue manager if a serious error occurs on a channel. Try to start the channel. If the problem continues, contact your MQSeries administrator.

If the status is RETRYING, then check the logs for the retry reason. Resolve the cause of the retry and check your application again. If the problem continues, start again at Step 1.

There is a list of common retry reasons at the end of this document. If the retry reason and resolution is not listed, contact your MQSeries administrator.
Note: Stopping and starting a channel in retry status does not help, you have to find the cause and resolve it.

The following steps, 5 - 9 are to be performed by Company B
Step 5:
Check your applications logs to ensure there are not any negative errors being returned by MQSeries. If there are errors, take note of the error. If the error points you directly to the cause of the problem, resolve the problem and check the application again. Otherwise, continue with Step 6.

Step 6:
Check your dead letter queue for messages. If any exist, check the reason. Resolve the reason. If there are not any messages in the dead letter queue related to this application, proceed to Step 7.

Note: The MCA can put message in the dead letter queue if it cannot deliver them to the destination queue successfully. Some examples of reasons are the destination queue is full or the queue name does not exist.

Checking for channel problems
Step 7:
Check the status of the receiver channel from Company A. Ask someone at Company A to tell you what the current status of their sender channel is. Verify your receiver channel of the same name has the same status. This seems like needless step, but it is possible to have strange MQSeries problems that can cause the status at each end of a channel to be different. If you find this, contact your MQSeries administrator.
MQSC> dis chs(companya.companyb)

Check that the application is servicing the request messages
Step 7:
Check the depth and the number of processes on the application queue. This is the queue the serving application gets the request messages from.
MQSC> dis ql(app.input) curdepth ipprocs

If the depth is greater than 0 and ipprocs is 0, your application is not servicing the queue. Check the application. Resolve the problem and check the entire application again. If the problem continues, check to see if the same problem occurred again or begin at Step 1.

If the depth is greater than 0 and ipprocs is greater than 0, your application is either not running properly or not servicing the messages fast enough. Check the application. Resolve the problem and check the entire application again. If the problem continues, check to see if the same problem occurred again or begin at Step 1.

If the depth is 0 or close to 0 and not rising, your application is servicing the messages. Go to Step 8.

Check to see if reply messages are being sent back to Company A.
Step 8:
Check the depth of the transmit queue that goes to Company A.
MQSC> Dis ql(companya) curdepth
If the depth is 0, check the last message date/time sent on your sender channel.
MQSC> dis chs(companyb.companya) lstmsgda lstmsgti
If the date/time is recent, your messages are being transmitted to Company A. **Skip to Step 10.**

If the date/time is not recent, your messages may be in the dead letter queue. You should have checked the dead letter queue in Step 6.

If the transmit queue depth is 0, your channel does not show a recent send date / time, and your messages are not in the dead letter queue, you most likely have an application problem.

If the depth is greater than 0, you most likely have a problem with the sender channel to Company A. **Proceed with Step 9**.

Step 9:
Check the status of your sender channel to Company A.
MQSC> Dis chs(companyb.companya)
If the status is STOPPED, try to determine why it is stopped. It can be stopped by a person or by the queue manager if a serious error occurs on a channel. Try to start the channel. If the problem continues, contact your MQSeries administrator.

If the status is RETRYING, then check the logs for the retry reason. Resolve the cause of the retry and check your application again. If the problem continues, start again at Step 1.

There is a list of common retry reasons at the end of this document. If the retry reason and resolution is not listed, contact your MQSeries administrator.
Note: Stopping and starting a channel in retry status does not help, you have to find the cause and resolve it.

The following step is to be performed by Company B
Step 10:
Check the depth and the number of processes on the reply queue. This is the queue the requesting application gets the reply messages from.
MQSC> dis ql(app.reply) curdepth ipprocs

If the depth is greater than 0 and ipprocs is 0, your application is not servicing the queue. Check the application. Resolve the problem and check the entire application again. If the problem continues, check to see if the same problem occurred again or begin at Step 1.

If the depth is greater than 0 and ipprocs is greater than 0, your application is either not running properly, not servicing the messages fast enough, or you have messages that did not return quickly enough **See Note.**. Check the application. Resolve the problem and check the entire application again. If the problem continues, check to see if the same problem occurred again or begin at Step 1.

Note: If your depth is greater than 0 and your application is servicing the queue, it is possible the reply messages are arriving on the reply queue after the application has stopped waiting for the reply. In this case you may have a process on the serving side not servicing the request quickly enough. You may have to increase the time your requesting application waits for a reply to resolve the problem or correct the problem on the serving side.

If the depth is 0 or close to 0 and not rising, your application is servicing the messages.

If you have come this far, your messages are making their full round trip and MQSeries is functioning fine. The problem is most likely in one of the applications or one of its supporting systems like a database.

COMMON CHANNEL RETRY REASONS
I am not going to detail resolution here today. Here are is a sample of reasons, not in exact MQ wording:
Sequence number error
Channel in doubt
Channel not found on other queue manager
No response from host