Exercise 12. Monitoring IBM MQ for performance

Estimated time

01:30

Overview

In this exercise, you enable and configure the online monitoring, statistics, and accounting features of IBM MQ.

Objectives

After completing this exercise, you should be able to:

- Enable accounting and statistics collection in IBM MQ
- View accounting and statistics data
- Configure a queue manager for online monitoring
- Monitor system resource usage

Introduction

With the IBM MQ monitoring facilities, you can postprocess monitoring data and monitor your queue manager online.

This exercise focuses on four types of monitoring:

1. **Statistics monitoring**
   Statistics monitoring generates data about the activity of the queue manager for a specified time interval.

   Statistics monitoring is used to collect information about the performance of various areas of the queue manager. You can collect data on the queue manager, queues, and channels, and you can configure the amount of information to collect.

   After it is collected, the data is made available at regular intervals in the form of PCF records that are written to the SYSTEM.ADMIN.STATISTICS.QUEUE system queue, with one message per subtype.

   This information can be used for charging purposes or for generating historical analysis of the operation or throughput of the IBM MQ system.

2. **Accounting monitoring**
   Accounting monitoring generates data about the activity of an IBM MQ connection.

Accounting monitoring collects information about each IBM MQ connection and when the connection disconnects. This information is written in the form of a PCF monitoring message to the SYSTEM.ADMIN.ACCOUNTING.QUEUE system queue. As an option, Interim messages can be written for long-running tasks.

As with statistics monitoring, the user can configure the amount of information that is collected. Separate messages can be written for each subtype of information that is collected.

This information can be used for charging or accounting purposes. The purpose of the online monitoring is to give a quick view of the performance of a specific resource. Data is immediately available, giving the user an instantaneous view of the particular resource. The monitoring data is collected for channels and queues, and is displayed as extra attributes on the **DISPLAY CHSTATUS** and the **DISPLAY QSTATUS** MQSC commands.

3. **Online real-time monitoring**
   Some queue and channel status attributes hold monitoring information, if real-time monitoring is enabled. You can enable or disable real-time monitoring for individual queues or channels, or for multiple queues or channels.
4. **System resource monitoring**
   Each queue manager publishes resource usage data to topics. This data is used by subscribers to those topics. When a queue manager starts, the queue manager publishes a set of messages on meta-topics. These messages describe which resource usage topics are supported by the queue manager, and the content of the messages published to those topics. Administrative tools can subscribe to the metadata to discover what resource usage information is available, and on what topics, and then subscribe to the advertised topics.

In this exercise, you configure and use the monitoring functions of IBM MQ. The exercise includes the following tasks:

1. Configure a queue manager to collect statistics data.
2. Configure a queue manager to collect accounting data.
3. Run test programs to generate statistics and accounting data.
4. Use the **amqsmon** sample program to display the information that is contained within accounting and statistics messages in a readable format. The **amqsmon** program reads accounting messages from the accounting queue, SYSTEM.ADMIN.ACCOUNTING.QUEUE, and reads statistics messages from the statistics queue, SYSTEM.ADMIN.STATISTICS.QUEUE.
5. Configure a queue manager for online monitoring and display the information.
6. Use the **amqsrua** command to enable and display MQ resource monitoring data. The **amqsrua** command reports metadata that queue managers publish. This data can include information about the CPU, memory, and disk usage.

Requirements

In addition, the following are required:

- IBM MQ and IBM MQ Explorer
- Successful completion of Exercise 4, “Connecting queue managers”
- The **amqsput**, **amqsget**, **amqsmon**, and **amqsrua** sample programs that are supplied with IBM MQ

- The **`statsdata.txt`** text file that is provided in the **`C:\labfiles\Lab12\data`** directory on Windows or the **`home/localuser/labfiles/Lab12/data`** directory on Linux

Exercise instructions

Part 1: Configure a queue manager to collect statistics data

In this part of the exercise, you configure a queue manager to collect statistics data. The configuration is controlled by using a set of queue manager and queue/channel attributes.

The attributes can be set by using the MQ Explorer or by using MQSC commands; instructions in this exercise use MQ Explorer.

__ 1. If it is not already running, start MQ Explorer.

__ 2. If it is not already running, start the queue manager that is named QM01.

__ 3. Edit the properties for queue manager QM01 to configure statistics monitoring properties.

__ a. In the **MQ Explorer - Navigator** view, right-click **QM01** and then click **Properties**.

__ b. To work with statistics properties, click **Statistics monitoring** in the **Properties** list.

Five queue manager properties control statistics generation:

○ **MQI statistics** (STATMQI) counts the number of MQ API requests processed in the recording interval.

○ **Queue statistics** (STATQ) records information about every local queue in the system. When a queue is accessed indirectly, for example by using a queue alias, the statistics are collected against the resolved local queue. Actions against remote queues are recorded against the associated transmission queue.

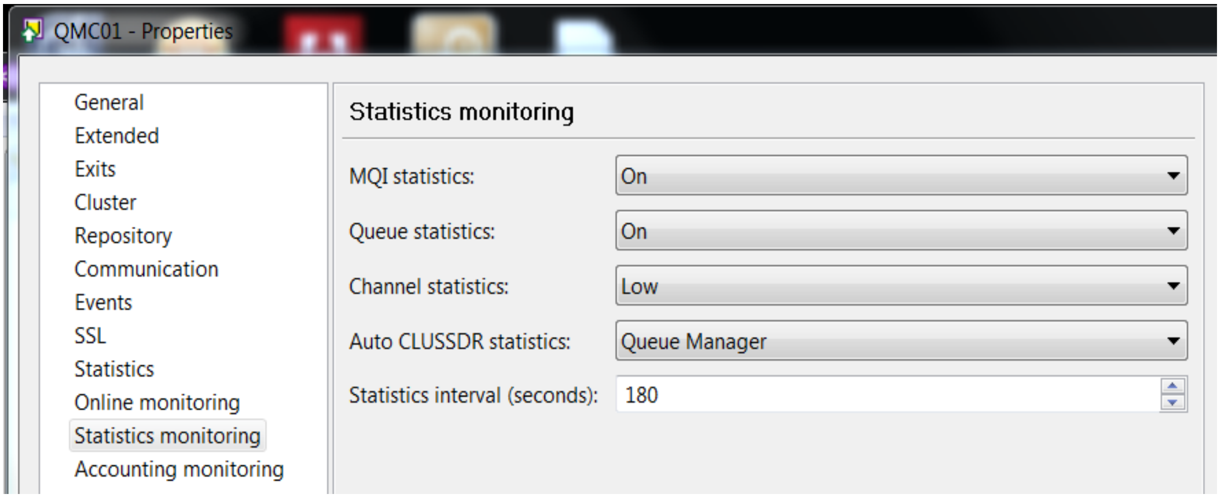
○ **Channel statistics** (STATCHL) records information about every channel (except SVRCONNs) in the system.

○ **Auto CLUSSDR statistics** (STATACLS) specifies whether to collect statistics data about the activity of auto-defined cluster-sender channels.

○ **Statistics interval** (STATINT) specifies, in seconds, the recording interval for recording data.

__ c. Activate statistics monitoring. In the **Statistics monitoring** window, set the following properties:

- MQI statistics: **On**
- Queue statistics: **On**
- Channel statistics: **Low**
- Statistics interval: **180**



Statistics monitoring

MQI statistics: On

Queue statistics: On

Channel statistics: Low

Auto CLUSSDR statistics: Queue Manager

Statistics interval (seconds): 180

With a **Statistics interval** of 180, statistics messages are generated every 3 minutes. The interval is set to a short interval to demonstrate this function. On production queue managers, the interval is likely to be set to a greater interval.

__ d. Click **Apply** and then click **OK**.

These settings enable statistics collection on the queue manager.

Note

You can disable statistics generation on each queue or channel by setting the **Queue statistics** (STATQ) or **Channel statistics** (STATCHL) attribute of the object to **Off**.

Part 2: Generate and view statistics data

In this step, you use the IBM MQ sample programs to:

- Put and get messages from a local queue
- Generate local queue statistics to put messages on a remote queue
- Generate statistics data for a channel

After the statistics are generated, you review the statistics data.

Note

This part of the exercises assumes that you successfully completed Exercise 4, “Connecting queue managers”.

__ 1. If it is not running, start the queue manager QMR01 and verify that the listener is running.

__ 2. On QMR01, define a local queue that is named **STATS.Q1**.

__ 3. On QM01, define a local queue that is named **STATS.Q1**.

__ 4. On QM01, define a local queue that is called **STATS.Q2**.

__ 5. On QM01, define a remote queue that is called **STATS.Q1.REMOTE** that points to the STATS.Q1 queue on queue manager QMR01.

Set the following remote queue definition properties:

- Remote queue: **STATS.Q1**
- Remote queue manager: **QMR01**
- Transmission queue: **QMR01**

__ 6. On QM01, verify that you have three queues on QM01 for statistics:

- A local queue that is named **STATS.Q1**
- A local queue that is named **STATS.Q2**
- A remote queue definition that is named **STATS.Q1.REMOTE**

__ 7. On QM01, start the sender channel QM01.QMR01.

Ensure that the channel is in the “Running” status before you continue.

Note

The QM01.QMR01 channel is the same channel that was used in Exercise 4, “Connecting queue managers”.

__ 8. Run the **`amqsput`** sample program to put messages to the remote queue definition on queue manager QM01 to send messages over the channel from QM01 to QMR01. Input messages to the queue by using **`statsdata.txt`** file, which contains 536 messages.

This command writes messages to the transmission queue (QMR01), which then is sent across the channel to queue manager QMR01.

W **Windows**

Type: **`amqsput STATS.Q1.REMOTE QM01 < C:\labfiles\Lab12\data\statsdata.txt`**

L **Linux**

Type:

```
amqsput STATS.Q1.REMOTE QM01 < /home/localuser/labfiles/Lab12/data/statsdata.txt
```

__ 9. Verify that you have 536 messages on STATS.Q1 on QMR01.

__ 10. Use the **amqsput** program to write messages to STATS.Q2 on QM01.

W **Windows**

Type: **`amqsput STATS.Q2 QM01 < C:\labfiles\Lab12\data\statsdata.txt`**

L **Linux**

Type: **`amqsput STATS.Q2 QM01 < /home/localuser/labfiles/Lab12/data/statsdata.txt`**

__ 11. Verify that you have 536 messages on STATS.Q2 on QM01.

__ 12. Use the **amqsget** sample program to read the messages off the STATS.Q2 queue.

In the command window, type:

```
amqsget STATS.Q2 QM01
```

__ 13. After the **amqsget** sample program ends, use the MQ sample program **amqsmon** to read the statistics messages from SYSTEM.ADMIN.STATISTICS.QUEUE on QM01.

Return to the window that you used to run **amqsmon**.

W **Windows**

Type: **`amqsmon -m QM01 -t statistics > c:\labfiles\Lab12\statistics.txt`**

L **Linux**

Type:

```
amqsmon -m QM01 -t statistics > /home/localuser/labfiles/Lab12/data/statistics.txt
```

__ 14. Open file **`statistics.txt`** in a text editor.

__ 15. The file contains several statistics messages that your tests generate. There are at least three statistics messages that are generated in each statistics interval (that is, as it is specified on the **Statistics Interval Queue Manager** property).

One message shows queue manager statistics. One message is generated for each queue to show queue statistics. One message is generated for the channel to show channel statistics.

Examine the file and locate the following messages:

- The queue manager statistics message for the test period
- The queue statistics message for STATS.Q2 for the test period
- The queue statistics message for QMR01 (that is, the transmission queue that is used to send messages to your other queue manager)
- The channel statistics message for QM01.QMR01 for the test period

__ 16. Disable statistics monitoring for QM01.

__ a. In the **MQ Explorer - Navigator** view, right-click **QM01** and then click **Properties**.

__ b. Click **Statistics monitoring** in the **Properties** list.

__ c. To deactivate statistics monitoring, set the following properties in **Statistics monitoring** window:

○ MQI statistics: **Off**
○ Queue statistics: **Off**
○ Channel statistics: **Off**

__ d. Click **Apply** and then click **OK**.

Part 3: Configure a queue manager to collect accounting data

In this part of the exercise, you configure a queue manager to collect accounting data. The configuration is controlled by using a set of queue manager, queue, and channel properties. These properties can be set by using MQ Explorer or MQSC commands; the instructions in this exercise are for MQ Explorer.

__ 1. In the **MQ Explorer - Navigator** view, right-click **QM01** and then click **Properties**. The properties for the QM01 queue manager are opened.

__ 2. To work with accounting properties, click **Accounting monitoring** in the **Properties** list.

Four queue manager properties control accounting data generation:

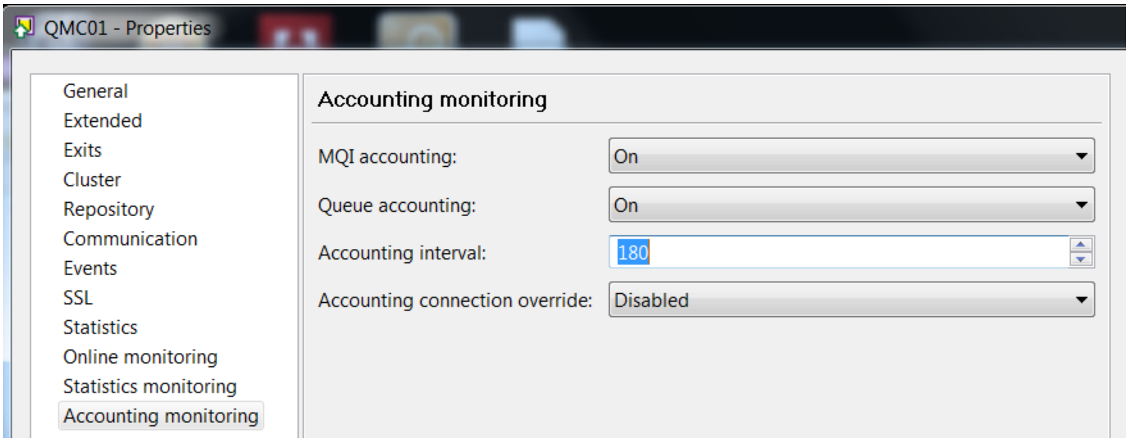
- **MQI accounting** (ACCTMQI) records collected information about the MQI operations that each application runs.
- **Queue accounting** (ACCTQ) records collected information about queues. Messages contain lists of records, one for each local queue open during the time interval.
- **Accounting interval** (ACCTINT) defines the interval in which to write intermediate records (that is, for long-running applications). Setting this attribute to zero turns off intermediate records.

- **Accounting connection override** (ACCTCONO) enables applications to override the attributes by using the **Connect** options in MQCONNX calls.

__ 3. Activate accounting data recording on QM1. In the **Accounting monitoring** properties window, set the following attributes:

- MQI accounting: **On**
- Queue accounting: **On**
- Accounting interval: **180**
- Accounting connection override: **Disabled**

__ 4. Click **Apply** and then click **OK**.



Accounting monitoring

MQI accounting: On

Queue accounting: On

Accounting interval: 180

Accounting connection override: Disabled

Part 4: Generate and view accounting data

In this part of the exercise, you run IBM MQ sample programs to simulate MQ activity and use the sample program to read and format the generated PCF messages. The test programs write messages to a local queue, and then read them back off the queue.

At the end of this part of the exercise, you review the generated accounting data.

__ 1. Create a local queue on QM01 that is named **ACCTG.STATS.Q1**.

__ 2. From a command window, run the **`amqsput`** program with the **`statsdata.txt`** file to put messages to the local queue, ACCTG.STATS.Q1. The **`statsdata.txt`** file is a text file of 536 lines, which generates a message for each line.

Windows

Type: **`amqsput ACCTG.STATS.Q1 QM01 < C:\labfiles\Lab12\data\statsdata.txt`**

Linux

Type:

```
amqsput ACCTG.STATS.Q1 QM01 </home/localuser/labfiles/Lab12/data/statsdata.txt
```

__ 3. Use MQ Explorer to check the queue depth of queue ACCTG.STATS.Q1 and verify that the queue contains 536 messages.

__ 4. Use the **amqsget** program to read messages from ACCTG.STATS.Q1.

In the command window, type:

```
amqsget ACCTG.STATS.Q1 QM01
```

This command gets all the messages off the queue.

__ 5. Check the queue depth of ACCTG.STATS.Q1. It should now be 0.

__ 6. Display the accounting data that is generated as a result of using **amqsput** and **amqsget**.

MQ provides a sample program that can be used to read and format statistics and accounting data.

Use the MQ sample program **amqsmon** to read the accounting PCF messages from SYSTEM.ADMIN.ACCOUNTING.QUEUE and write the results to text file.

W **Windows**

Type: `amqsmon -m QM01 -t accounting > c:\labfiles\Lab12\accounting.txt`

L **Linux**

Type:

```
amqsmon -m QM01 -t accounting > /home/localuser/labfiles/Lab12/accounting.txt
```

__ 7. Open the file **accounting.txt** in a text editor.

The file contains several messages that the **amqsput** and **amqsget** sample programs generate when you use them.

Look at the file and locate the messages that are associated with **amqsput** and **amqsget**. Each program (**amqsput** and **amqsget**) generates two messages. One message is generated to show MQI accounting, that is, the MQI operations that get each application runs. The other message is generated to show queue accounting, that is, the details of the queue that the application uses.

Note

For applications that use multiple queues, a separate message exists for each queue.

Look at the file and locate the MQI accounting and queue accounting messages for each application.

Table 1 and Table 2 describe the fields in each message.

Table 1. MQI accounting fields

Field	Description
QueueManager	Name of the queue manager.
IntervalStartDate	Date of the start of the monitoring period.
IntervalStartTime	Time of the start of the monitoring period.
IntervalEndDate	Date of the end of the monitoring period.
IntervalEndTime	Time of the end of the monitoring period.
CommandLevel	IBM MQ command level.
ConnectionId	Connection ID for the task.
SeqNumber	Sequence number (normally zero). Increments for each subsequent record for long running connections.
ApplicationName	Name of the application (program name).
ApplicationPid	Operating system process ID of the application.
ApplicationTid	IBM MQ thread ID of the connection in the application.
User ID	User ID context of the application.
ConnDate	Normally matches “StartTime,” except for long-running connections.
ConnTime	Normally matches “StartTime,” except for long-running connection.
ConnName	Only written for client connections.
ChannelName	Only written for client connections.
DiscDate	Normally matches “EndTime,” except for long running connections.
DiscTime	Normally matches “EndTime,” except for long running connections.
DiscType	MQ_DISCONNECT_TYPE_NORMAL, application requested. MQ_DISCONNECT_TYPE_IMPLICIT, abnormal application termination. MQ_DISCONNECT_TYPE_QMGR, queue manager broke the connection (`shutdown -i`).
OpenCount	Number of objects opened.
OpenFailCount	Number of unsuccessful open object attempts.
CloseCount	Number of objects closed.
CloseFailCount	Number of unsuccessful close object attempts.
PutCount	Number of messages that are successfully put to a queue (excluding MQPUT1 requests).
PutFailCount	Number of unsuccessful put message attempts.
PutBytes	Number bytes written in MQPUTs.
GetCount	Number of successful destructive MQGETs.

Table 1. MQI accounting fields

Field	Description
GetFailCount	Number of unsuccessful destructive gets.
GetBytes	Total number of bytes retrieved.
BrowseCount	Number of successful non-destructive gets.
BrowseFailCount	Number of successful non-destructive gets.
BrowseBytes	Total number of bytes browsed.
CommitCount	Number of successful transactions that are completed including those transactions that the application commits when it disconnects.
CommitFailCount	Number of unsuccessful attempts to complete a transaction that include those transactions that run an implicit commit because the application disconnected.
BackCount	Number of backouts processed (including implicit backout upon abnormal disconnect).
InqCount	Number of successful objects inquired upon.
InqFailCount	Number of unsuccessful “object inquire” attempts.
SetCount	Number of successful object updates (SET).
SetFailCount	Number of successful SET attempts.

Note: The OpenCount, OpenFailCount, CloseCount, and CloseFailCount fields are displayed as arrays.

Table 2. Queue accounting fields

QueueManager	Name of the queue manager
IntervalStartDate	Date of the start of the monitoring period.
IntervalStartTime	Time of the start of the monitoring period.
IntervalEndDate	Date of the end of the monitoring period.
IntervalEndTime	Time of the end of the monitoring period.
CommandLevel	IBM MQ command level.
ConnectionId	Connection ID for the task.
SeqNumber	Sequence number normally zero. Increments for each subsequent record for long running connections.
ApplicationName	Name of the application (program name).
ApplicationPid	Operating system process ID of the application.
ApplicationTid	IBM MQ thread ID of the connection in the application.
User ID	User ID context of the application.
ObjectCount	Number of queues that are accessed in the interval for which accounting data were recorded.
QueueName	Name of the queue.

Table 2. Queue accounting fields

QueueManager	Name of the queue manager
CreateDate	Date the queue was created.
CreateTime	Time the queue was created.
QueueType	Type of the queue.
QueueDefinitionType	Queue definition type. Valid values are: • PREDEFINED • PERMANENT_DYNAMIC • TEMPORARY_DYNAMIC
OpenCount	Number of times the application opened this queue in this interval.
OpenDate	Date of the first open of the queue in this recording interval. If the queue was already open at the start of this interval, this value is the date that the queue was originally opened.
OpenTime	Time of first open of the queue in this recording interval. If the queue was already open at the start of this interval, this value is the time that the queue was originally opened.
CloseCount	Number of objects closed.
CloseDate	Date of final close of the queue in this recording interval. If the queue is still open, then the value is not returned.
CloseTime	Time of final close of the queue in this recording interval. If the queue is still open, then the value is not returned.
PutCount	Number of messages that are successfully put to the queue (excluding MQPUT1 requests).
PutFailCount	Number of unsuccessfully put message attempts.
Put1Count	Number of messages that are successfully put to the queue through MQPUT1 requests.
Put1FailCount	Number of unsuccessful MQPUT1 message attempts.
PutBytes	Total number of bytes put.
PutMinBytes	Smallest message size that is placed on the queue.
PutMaxBytes	Largest message size that is placed on the queue.
GetCount	Number of successful destructive MQGETs.
GetFailCount	Number of unsuccessful destructive GETs.
GetBytes	Number of bytes read in destructive GETs.
GetMinBytes	Size of the smallest messages that are retrieved from the queue.
GetMaxBytes	Size of the largest messages that are retrieved from the queue.
BrowseCount	Number of successful non-destructive GETs.
BrowseFailCount	Number of unsuccessful non-destructive MQGETs.

Table 2. Queue accounting fields

QueueManager	Name of the queue manager
BrowseBytes	Number of bytes read in non-destructive MQGETs that returned persistent messages.
BrowseMinBytes	Size of the smallest messages that are browsed from the queue.
BrowseMaxBytes	Size of the largest messages that are browsed from the queue.
GeneratedMsgCount	Number of messages generated (report / event).
TimeOnQMin	Shortest time a message remained on the queue before it is retrieved (in microseconds).
TimeOnQAvg	Average time a message remained on the queue before it is retrieved (in microseconds).
TimeOnQMax	Longest time a message remained on the queue before it is retrieved (in microseconds).

__ 8. Disable accounting monitoring on the QM01 queue manager.

__ a. In the **MQ Explorer - Navigator** view, right-click **QM01** and then click **Properties**. The properties for the QM01 queue manager are opened.

__ b. Click **Accounting monitoring** in the **Properties** list.

__ c. Deactivate accounting data recording on QM1. In the **Accounting monitoring** properties window, set the following attributes:

- ○ MQI accounting: **Off**
- ○ Queue accounting: **Off**

__ d. Click **Apply** and then click **OK**.

Part 5: Configure a queue manager for online monitoring

In this section, you configure a queue manager for online monitoring. By default, online monitoring data is not collected. It must be enabled by using a set of queue manager attributes.

The configuration of monitoring data collection has three values to enable the collection of data (LOW, MEDIUM, and HIGH). Collecting monitoring data might require that you run some instructions that affect performance (for example, obtaining system time).To reduce the impact of online monitoring, the resource-intensive measurements are sampled instead of measuring every value.

The online monitoring values can be set on each queue manager. The configuration is controlled by using a set of queue manager, queue, and channel attributes. These values can be set by using the MQ Explorer or MQSC commands; the instructions in this exercise are for MQ Explorer.

__ 1. In MQ Explorer, open the properties for queue manager QM01.

__ 2. Select **Online monitoring**.

Three queue manager properties control online monitoring:

- **Channel monitoring** (MONCHL) controls the collection of online monitoring data for channels.
- **Queue monitoring** (MONQ) controls the collection of online monitoring data for queues.
- **Auto CLUSSDR monitoring** (MONACLS) controls the collection of monitoring data for cluster sender channels.

__ 3. Activate recording of statistics for channels and queues. In the **Online Monitoring** properties for QM01, set the following values:

- Channel monitoring: **High**
- Queue monitoring: **High**

__ 4. Run the **`amqsput`** sample program with the **`statsdata.txt`** file to put 536 messages on STATS.Q2.

__ 5. Run the **`amqsget`** sample program against STATS.Q2 to get the messages.

__ 6. Review the generated online monitoring data for QM01 by using MQSC commands.

In the MQSC session for QM01, type:

```
DIS QSTATUS(STATS.Q2) MONITOR
```

This command displays the monitoring data for the queue. Table 3 summarizes the monitoring fields.

Table 3.

QTIME	The interval between when messages are put on the queue and when they are destructively read. The value is shown in microseconds, formatted into a string of up to 9 digits.
LPUTDATE	Date on which the last message was put to the queue since the queue manager started. If no PUT date is available, then the value is blank.
LPUTTIME	Time at which the last message was put to the queue since the queue manager start. If no put time is available, the value shows as blank.
LGETDATE	Date on which the last message was retrieved from the queue since the queue manager start. A message that is browsed does not count as a retrieved message. If no get date is available, the value is shown as blank.
LGETTIME	Time at which the last message was retrieved from the queue since the queue manager start. A browsed message does not count as a retrieved message. If no get time is available, the value is shown as blank.
MONQ	Current level of monitoring data collection for the queue. Possible values: OFF, LOW, MEDIUM, HIGH.
MSGAGE	Age, in seconds, of the oldest message on the queue. The maximum displayable value is 999 999 999. If the age exceeds this value, 999 999 999 is displayed.

__ 7. Disable online monitoring on the QM01 queue manager.

__ a. In MQ Explorer, open the properties for queue manager QM01.

__ b. Click **Online monitoring**.

__ c. Deactivate recording of statistics for channels and queues. In the **Online Monitoring** properties for QM01, set the following values:

- Channel monitoring: **Off**
- Queue monitoring: **Off**

__ d. Click **Apply** and then click **OK**.

Part 6: Resource monitoring

In this part of the exercise, you use the **amqsrua** sample program to enable and display MQ resource monitoring data. The **amqsrua** command reports metadata that is published by queue managers. This data can include information about the CPU, memory, and disk usage.

In this part of the exercise, you need two command windows (sessions). In one window, you run the **amqsrua** sample program. In the second window, you run the commands to put and get messages to a queue on a QM01

__ 1. Use the **amqsrua** sample program in interactive mode to get a queue status summary.

__ a. In one of the command windows, type:

```
amqsrua -m QM01
```

__ b. The command responds with a set of resource classes: CPU, DISK, STATMQI, and STATQ.

Type **STATQ** and then press Enter.

__ c. You can now specify the type of action you want to monitor for a queue: OPENCLOSE, INQSET, PUT, or GET.

Type **PUT** and then press Enter.

__ d. Specify the name of the queue to monitor.

Type **STATS.Q1** and then press Enter.

The **amqsrua** sample program is now subscribing to the specified class of resource monitoring events. Using the command in this way reports on the values of those resource elements every 10 seconds. Currently, the queue is empty so all values are zero.

__ 2. In the second command window, run the **amqsput** program to put messages to the STATS.Q1 queue on QM01.

W **Windows**

Type: **`amqsput STATS.Q1 QM01 < C:\labfiles\Lab12\data\statsdata.txt`**

L **Linux**

Type: **`amqsput STATS.Q1 QM01 < /home/localuser/labfiles/Lab12/data/statsdata.txt`**

__ 3. Observe the counts in queue status window. You should see that the MQPUT non-persistent message count is now 536.

__ 4. End the **amqsrua** program. Type: Ctrl+C

__ 5. Run the **amqsrua** program in interactive mode to get the log statistics.

__ a. In the command windows, type:

```
amqsrua -m QM01
```

__ b. For the class selection, type **DISK** and then press Enter.

__ c. For the type selection, type **Log** and then press Enter.

The Log statistics include the bytes in use, the maximum bytes available, physical bytes written, logical bytes written, and write latency.

The command should return output that is similar to the following example:

```
Publication received PutDate: 20161027 PutTime: 19064332 Interval: 10.00
seconds
Log - bytes in use 50331648
Log - bytes max 83886080
Log - physical bytes written 4096 19/sec
Log - logical bytes written 1352 6/sec
Log - write latency 15539 uSec
```

__ d. End the **amqsrua** program.

__ 6. Run the **amqsrua** program in command-line mode to get message data on directories for errors, trace, and the number of MQ FDC records. Type:

```
amqsrua -m QM01 -c DISK -t SystemSummary
```

The command should return output similar to the following example:

```
Publication received PutDate: 20161027 PutTime: 19064332 Interval: 10.00
seconds
MQ errors file system - bytes in use 204521MB
MQ errors file system - free space 57.11%
MQ FDC file count 4
MQ trace file system - bytes in use 204521MB
MQ trace file system - free space 57.11%
```

__ 7. The **amqsrua** sample program subscribes to the MQ resource monitoring topics. Verify the subscription and the topic string in MQ Explorer.

__ a. Click **Subscriptions** under QM01 in the **MQ Explorer - Navigator** view.

__ b. You should see one subscription for the topic string:
$SYS/MQ/INFO/QMGR/QM01/Monitor/DISK/SystemSummary

Information

You can run the **amqsrua** program with the argument **-d 1** to enable debug mode. The results show the system topics that the program subscribes to.

__ 8. End the **amqsrua** program.

End of exercise

Exercise review and wrap-up

You should now be able to:

- Use the IBM MQ Explorer to enable accounting, statistics, and online monitoring collection
- View accounting, statistics, and online monitoring data
- Use the **`amqsrua`** sample program to subscribe to MQ resource monitoring topics

IBM Training