

Unit 13. Monitoring and configuring IBM MQ for performance

Estimated time

01:30

Overview

In this unit, you learn about the information that the accounting and statistics system management utilities provide for monitoring an IBM MQ network.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM Knowledge Center for IBM MQ V9

Unit objectives

- Describe the statistics and accounting data that IBM MQ provides
- View and generate accounting and statistical data
- Subscribe to IBM MQ statistic topics
- Interpret statistics and accounting data to identify possible system performance benefits
- Configure and tune IBM MQ for improved performance

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-1. Unit objectives

Collecting statistics and accounting data

- IBM MQ collects sets of data to be written as messages to predefined queues
 - Data can be post-processed to give information about system activity
 - Data can be used for capacity planning, chargeback, and other information
- Statistical data collection is divided into three categories:
 - MQI statistics
 - Queue statistics
 - Channel statistics
- Collection of data for each class can be selected independently
- Queue manager and queue or channel attributes control collection
- Accounting data collects information about MQI applications
- Can be activated by modifying MQ object properties or by subscribing to IBM MQ statistics topics

Figure 13-2. Collecting statistics and accounting data

MQ can generate statistics messages that record the information about the activities that occur in an MQ system.

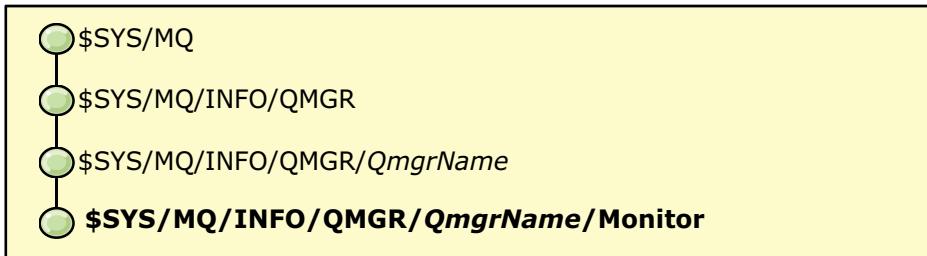
The information that is contained within statistics messages can be used for:

- Accounting for application resource use
- Recording application activity
- Planning for capacity
- Detecting problems in your queue manager network
- Helping to determine the causes of problems in your queue manager network
- Improving the efficiency of your queue manager network
- Familiarizing yourself with the running of your queue manager network
- Confirming that your queue manager network is running correctly

Statistical data is available for MQI, queues, and channels.

Subscribing to statistics data (1 of 2)

- IBM MQ publishes resource monitoring data from the queue manager to \$SYS/MQ topic branch
 - Access to \$SYS/MQ is restricted to administrators by default
 - Others can subscribe to a subset of the data
- Not controlled by queue manager configuration options
 - Statistics topic data is generated only when an application subscribes to the system topic
- Statistics are organized according to class and type: CPU, DISK, STATMQI, STATQ
- Includes all statistics in SYSTEM.ADMIN.STATISTICS.QUEUE plus statistics for monitoring CPU and disk



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-3. Subscribing to statistics data (1 of 2)

IBM MQ Version 9.0 publishes statistics and activity trace information messages to MQ system level topic strings.

In MQ, statistics are published to a system topic under \$SYS/MQ/INFO/QMGR. An authorized user can subscribe to the topics to receive monitoring information for the queue manager.

You must be authorized at, or deeper than, \$SYS/MQ to be granted authority to use the \$SYS/MQ topic tree.

You can use MQSC and IBM MQ Explorer to define and monitor topics and subscriptions.

Subscribing to statistics data (2 of 2)

- Statistics are published approximately every 10 seconds
- STATINT queue manager attribute controls the interval over which the subscription high and low watermarks are taken
- IBM MQ sample program `amqsrua` reports metadata from queue manager topic
 - Source file is in the IBM MQ `samples` directory
- User applications discover which statistics are available by subscribing to a system topic
- Topic tree structure:
`$SYS/MQ/INFO/QMGR/<QmgrName>/Monitor/<Class>/<Type>`
Example: `$SYS/MQ/INFO/QMGR/QMGR1/Monitor/CPU/QMgrSummary`

Benefits of subscribing to statistics topics

- Dynamically enable and disable statistics without modifying queue manager configuration
- Supports multiple subscribers to the same set of information, allowing more than one monitoring tool to access statistics
- Ability to give non-administrative users permission to subscribe to a subset of information specific to their application resources

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-5. Benefits of subscribing to statistics topics

IBM MQ statistics that are published to topics only

- CPU statistics
 - User CPU time
 - System CPU time
 - CPU load
 - RAM free percentage CPU
 - RAM total bytes
- STATQ PUT statistics
 - Lock contention
 - Queue avoided puts
 - Queue avoided bytes
- DISK statistics
 - Log bytes maximum
 - Log physical bytes written
 - Log logical bytes written
 - Log write latency

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-6. IBM MQ statistics that are published to topics only

This figure summarizes the MQ statistics that are available only in published topics.

- **User CPU time** is the average (taken over the last 10-second interval) percentage of time that is used by the CPU when it was in non-privileged code.
- **System CPU time** is the average (taken over the last 10-second interval) percentage of time that is used by the CPU when it was in privileged code.
- **CPU load** is the load average.
- **RAM total bytes** is an approximation of the memory that is used by the queue manager.
- **Log bytes maximum** refers to the maximum number of bytes that can be written to the log when all the primary and secondary extents are full. This value is less than the size of the log file system.
- **Log physical bytes written** and **Log logical bytes written** are the physical number of bytes written to disk.
- **Log write latency** is a rolling average that represents the time that a single write to disk takes.
- **Lock contention** is the percentage of attempts to lock the queue that resulted in waiting for another process to release the lock first. Decreasing lock contention is likely to increase the

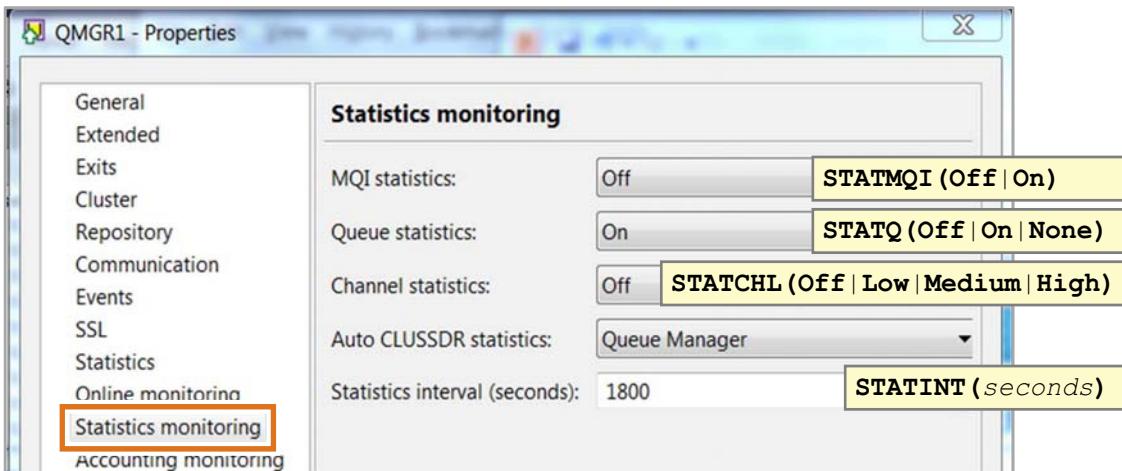
maximum throughput of your system because taking a lock that is not currently locked is a much cheaper operation than waiting for a lock to be released.

- If a message is put to a queue when a “getter” is waiting, the message might not need to be queued as it might be possible for it to be passed to the getter immediately. It is said that this message avoided the queue. **Queue avoided puts** and **Queue avoided bytes** are the count of such messages and bytes. Increasing queue avoidance is likely to increase the maximum throughput of your system because it avoids the cost of putting the message onto the queue and getting it off again.

For more information, see the “Statistics published to the system topic in MQ v9” MQdev Blog on IBM DeveloperWorks at:

https://www.ibm.com/developerworks/community/blogs/messaging/entry/Statistics_published_to_the_system_topic_in_MQ_v9?lang=en

Queue manager statistics monitoring properties



- Edit properties on this page to configure how statistics monitoring data is collected for queues and channels on this queue manager

Figure 13-7. Queue manager statistics monitoring properties

You can enable statistics collection for MQI, queues, and channels on the queue manager by using the MQSC commands or the queue manager **Statistics monitoring** properties in IBM MQ Explorer. The figure shows the IBM MQ Explorer statistics property and associated MQSC commands.

The queue manager MQI statistics property (**STATMQI**) controls the collection of MQI statistics. When the MQI statistics property is set to **On**, the MQI statistics information is collected for every connection to the queue manager.

Queue statistics can be enabled for individual queues, or for multiple queues at the queue manager level, by using the queue manager **Queue statistics** property (**STATQ**). When **Queue statistics** is set to **On**, queue statistics are collected for queues that have the **Queue statistics** property set to **Queue Manager**.

The queue manager **Channel statistics** property (**STATCHL**) controls the collection of channel statistics for all channels, or individual channels.

The **Statistics interval** property (**STATINT**) specifies the interval, in seconds, between the generation of statistics messages. The default statistics interval is 1800 seconds (30 minutes).

MQI statistics

- Generated only for queues that are opened after statistics collection is enabled
- Contain information about the number of MQI calls made during a configured interval
 - Success and failure counts for each MQI verb
 - MQI and byte counts for PUT and GET on queues for persistent and nonpersistent messages
- Written in the form of PCF records to SYSTEM.ADMIN.STATISTICS.QUEUE and IBM MQ topics

- Methods for enabling MQI statistics:
 - Use the MQSC command **ALTER QMGR** with **STATMQI(ON)**
 - Subscribe to **STATMQI** statistics topic
 - Use IBM MQ Explorer to modify queue manager **Statistics Monitoring** and queue **Statistics** properties

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-8. MQI statistics

MQI statistics must be enabled on a queue before it is opened.

MQI statistics messages contain information that relates to the number of MQI requests processed during a configured interval. For example, the statistics message information can include the number of MQI commands that a queue manager processes.

A statistics message is a PCF message that contains PCF structures. Statistics messages are delivered to the system queue (SYSTEM.ADMIN.STATISTICS.QUEUE) at configured intervals.

You can use MQSC and IBM MQ Explorer to control statistics generation. In IBM MQ V9, you can also subscribe to MQ topics to activate statistics collection.

Queue statistics

- Can be configured at both queue manager and queue level
- Minimum and maximum depth of queue
- Average time-on-queue for messages that are retrieved from the queue
- API and byte counts for GET, PUT, BROWSE (persistent and non-persistent)
- Methods for enabling queue statistics:
 - Alter the queue definition to use the **STATQ(ON)** attribute
 - Subscribe to **STATQ** statistics topic
 - Use IBM MQ Explorer to modify queue manager **Statistics Monitoring** and queue **Statistics** properties

Figure 13-9. Queue statistics

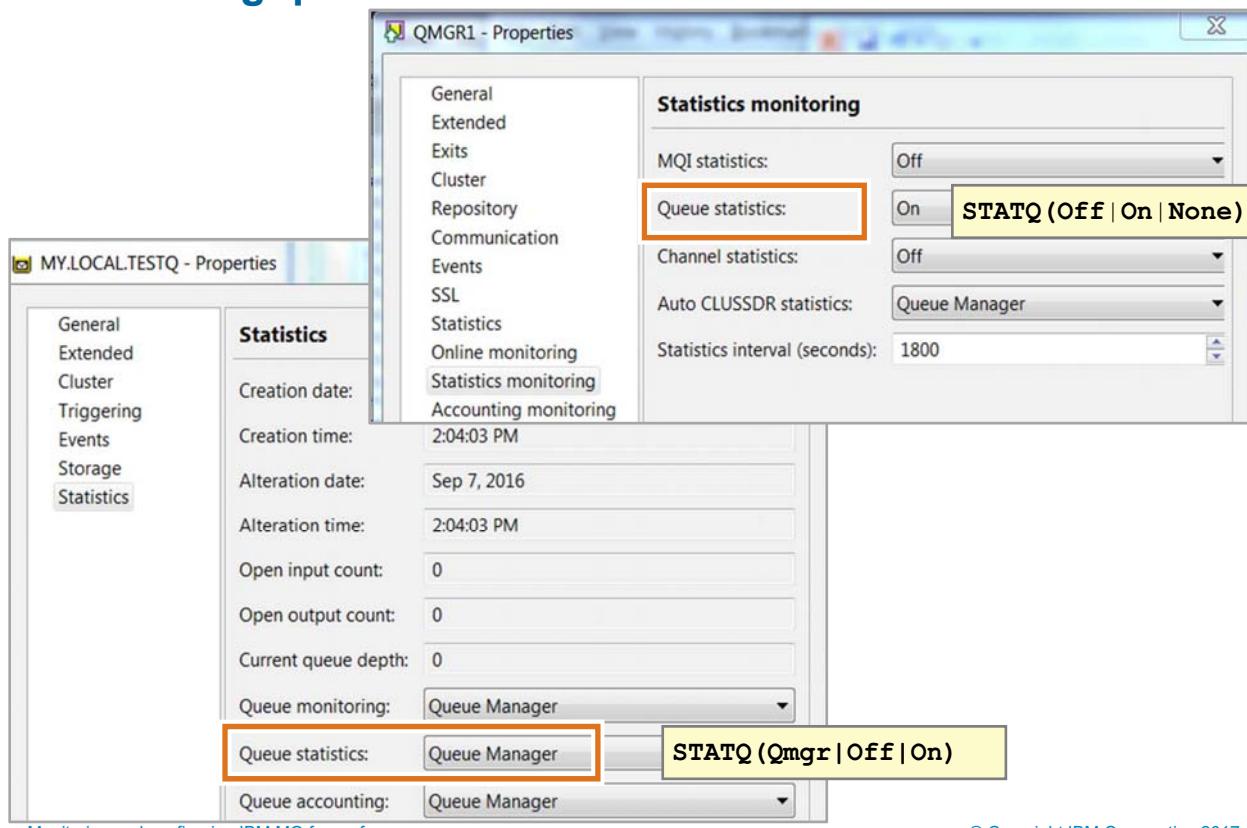
Queue statistics messages contain information about the activity of a queue during a configured interval. The information includes the number of messages that are put on and retrieved from the queue, and the total number of bytes that a queue processes.

Each queue statistics message can contain up to 100 records. Each record contains activity information for each queue for which statistics were collected.

You can enable queue statistics by using MQSC, subscribing to the MQ STATQ topic, or by using IBM MQ Explorer.

IBM Training

Controlling queue statistics



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-10. Controlling queue statistics

By using MQ Explorer or MQSC commands, you can enable queue statistics at the queue manager level or at the queue.

If the **Queue statistics** property is set to **Queue Manager** on the queue, the queue manager controls statistics collection. If the **Queue statistics** properties are set to **On** on the queue, statistics information is collected for every connection to the queue manager that opens the queue.

When you use MQSC, the queue attribute **STATQ** on the **ALTER** command for the queue controls individual channels. The queue manager attribute **STATQ** on the **ALTER OMGR** command controls many queues together.

Channel statistics

- Messages are written at end of the interval that contains up to 100 channel records
- Only channels “active” in the time interval have statistics that are recorded
- Can be configured at both queue manager and channel levels by using MQSC or IBM MQ Explorer
 - Set **STATACLS** queue manager attribute to control automatically defined cluster-sender channels

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-11. Channel statistics

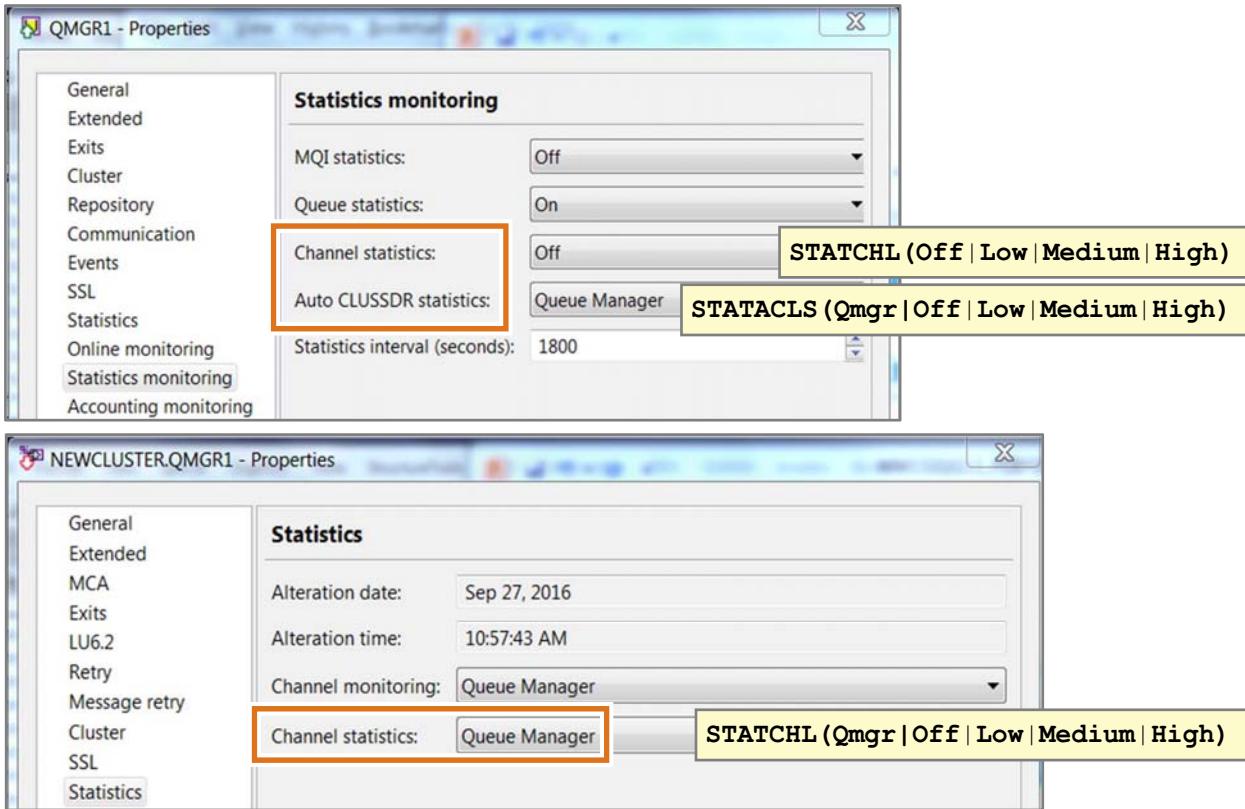
Channel statistics messages contain information about the activity of a channel during a configured interval. For example, the message can include the number of messages that the channel transferred, or the number of bytes that the channel transferred.

Each channel statistics message contains up to 100 records. Each record contains the activity for each channel for which statistics were collected.

Channel statistics information collection can be set to one of the three monitoring levels: low, medium, or high. Collecting statistics information data might require that the process runs some instructions that can affect performance. To reduce the impact of channel statistics collection, the “medium” and “low” monitoring options measure a sample of the data at regular intervals rather than collecting data all the time.

Automatically defined cluster-sender channels are not MQ objects, so they do not have attributes. To control automatically defined cluster-sender channels, use the queue manager attribute **STATACLS**. This attribute determines whether automatically defined cluster-sender channels within a queue manager are enabled or disabled for channel statistics information collection.

Controlling channel statistics



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-12. Controlling channel statistics

By using MQ Explorer or MQSC commands, you can enable queue and channel statistics at the queue manager level or at the individual queue or channel level.

If the **Channel statistics** property on the channel is set to **Queue Manager**, the queue manager controls statistics collection. If the **Channel statistics** property on the channel is set to **Off**, no statistics information is collected for this channel.

When you use MQSC, the channel attribute **STATCHL** on the **ALTER CHL** command controls individual channels. The queue manager attribute **STATCHL** on the **ALTER QMGR** command controls many channels together.

IBM MQ accounting data collection

- Can collect connection-level and queue information for each connection
 - MQI accounting data that is controlled by queue manager
 - Queue accounting that is controlled by queue manager and queue
 - Accounting interval in seconds
 - Applications can override the MQI accounting attribute and the Queue accounting attribute by using the Connect options in MQCONN calls
- Connection-level information includes:
 - Context details such as application name, process ID, connection type, and connect time
 - MQI counts
 - Message details (counts, bytes) for MQPUT, MQGET, MQGET (browse)
- Written in the form of a PCF monitoring message to SYSTEM.ADMIN.ACCTING.QUEUE

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-13. IBM MQ accounting data collection

Accounting messages record information about the MQI operations that MQ applications complete. An accounting message is a PCF message that contains a number of PCF structures.

When an application disconnects from a queue manager, an accounting message is generated and delivered to the system accounting queue SYSTEM.ADMIN.ACCTING.QUEUE. For long-running MQ applications, intermediate accounting messages are generated as follows:

- When the time since the connection was established exceeds the configured interval
- When the time since the last intermediate accounting message exceeds the configured interval

The information that is contained within accounting messages can be used for the following purposes:

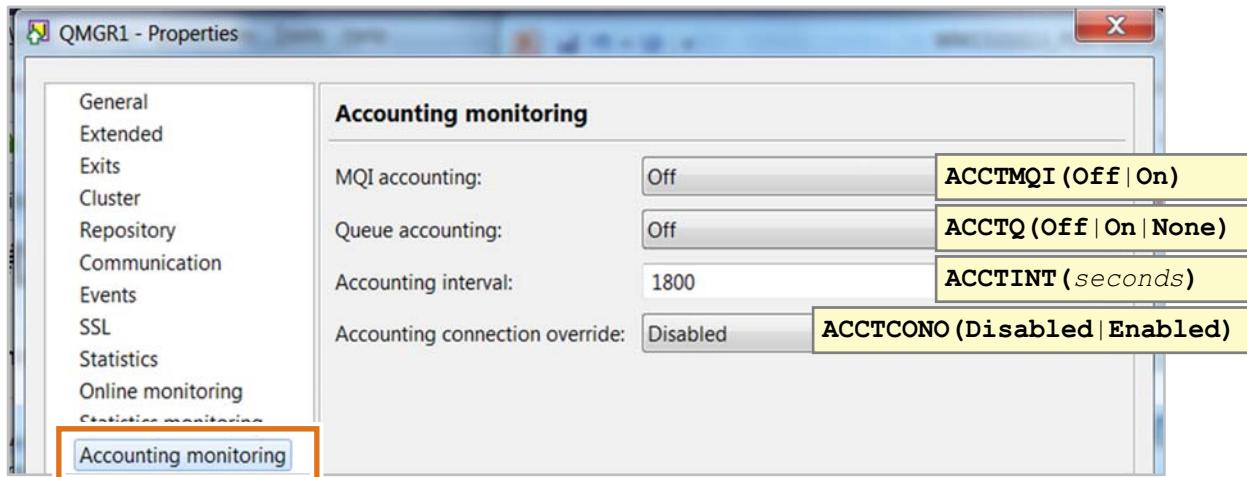
- Accounting for application resource use
- Recording application activity
- Detecting problems in your queue manager network
- Helping to determine the causes of problems in your queue manager network
- Improving the efficiency of your queue manager network
- Familiarizing yourself with the running of your queue manager network
- Confirming that your queue manager network is running correctly

MQI accounting messages contain information that is related to the number of MQI requests that were run by using a connection to a queue manager.

The application can also modify the collection of both MQI and queue statistics at the connection level by specifying the `ConnectOpts` parameter on the MQCONN call.



Controlling queue manager accounting



- Can set properties for both local and remote queue managers

Figure 13-14. Controlling queue manager accounting

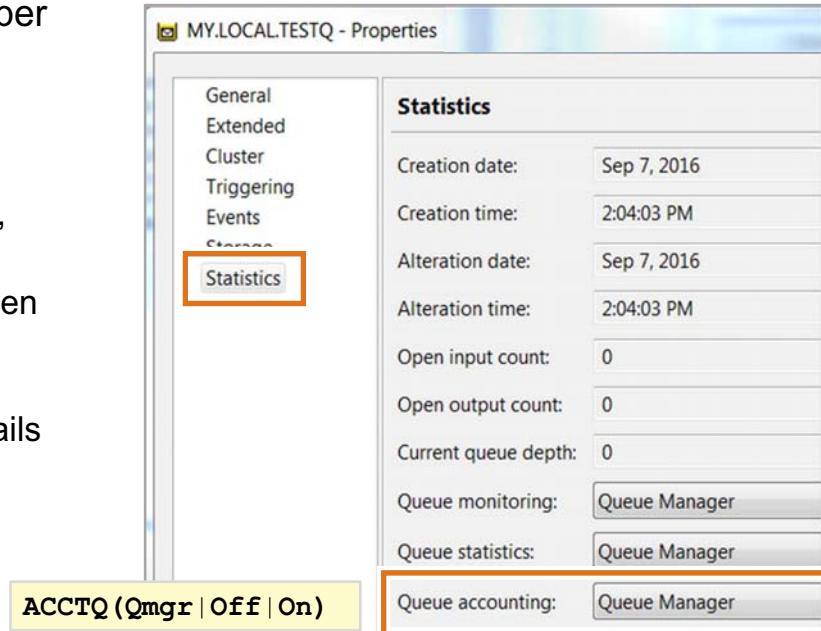
You can enable MQI and queue accounting on the queue manager by using MQSC commands or the queue manager **Accounting monitoring** properties in MQ Explorer. The figure shows the MQ Explorer **Accounting monitoring** properties and the associated MQSC command.

The queue manager **MQI accounting** property (**ACCTMQI**) controls the collection of MQI accounting information. When **MQI accounting** is set to **On**, accounting information is collected for every connection to the queue manager.

The queue manager **Queue accounting** property (**ACCTQ**) controls the collection of queue accounting information for any queues with the **Queue accounting** property set to **Queue Manager**.

Controlling queue accounting

- Queue data might be up to 100 queue details per message
- Queue information includes:
 - Queue details: Name, type
 - Open details: First open time, last close time
 - MQPUT, MQGET, MQGET(browse) details
- Control at queue manager level or at queue level



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-15. Controlling queue accounting

Queue accounting messages contain information about the number of MQI requests that were processed by using connections to a queue manager, concerning specific queues.

Each queue accounting message can contain up to 100 records. Every record contains information about application activity for a specific queue.

Displaying statistics and accounting data

- Use `amqsmon` sample program to display the information that is contained within accounting and statistics messages in a formatted form
 - Accounting messages are read from the accounting queue, `SYSTEM.ADMIN.ACCTOUNTING.QUEUE`
 - Statistics messages are read from the statistics queue, `SYSTEM.ADMIN.STATISTICS.QUEUE`
- Source code is provided
- Examples:

```
amqsmon -m QMGR1 -t statistics
amqsmon -m QMGR1 -t accounting
```

Figure 13-16. Displaying statistics and accounting data

Accounting and statistics messages are written to the system accounting and statistics queues. To use the information that is recorded in these messages, you must use an application to transform the recorded information into a format that is suitable for use.

The MQ sample program, `amqsmon`, processes messages from the accounting and statistics queues and shows the information in a readable format.

The supplied source code for the `amqsmon` program can be used as a template for writing your own application to process accounting or statistics messages. You can also modify the `amqsmon` source code to meet your own particular requirements.

The required parameter is the type (`-t`) of messages to process:

- If `-t` is set to `accounting`, accounting record messages are processed from the system queue `SYSTEM.ADMIN.ACCTOUNTING.QUEUE`
- If `-t` is set to `statistics`, statistics records are processed. Messages are read from the system queue `SYSTEM.ADMIN.STATISTICS.QUEUE`

For a complete description the `amqsmon` parameters, see the IBM MQ product documentation.

Examples of accounting data

```
QueueManager: QMGR1
IntervalEndDate: 2016-08-10
IntervalEndTime: 14:39:50
CommandLevel: 900
SeqNumber: 0
ApplName: amqsput.exe
ApplicationPid: 9408
ApplicationTid: 1
UserId: 'admin01'
ObjectCount: 1
```

```
OBJECTS:
QueueName: 'APP.QUEUE.X'
QueueType: Predefined
QueueDefType: Local
OpenCount: 1
OpenDate: 2016-08-10
OpenTime: 14:39:49
CloseCount: 1
CloseDate: 2016-08-10
CloseTime: 14:39:50
```

```
PutCount: [0, 1]*
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [0, 4]
PutMinBytes: [0, 4]
PutMaxBytes: [0, 4]
*array shows nonpersistent, persistent
```

Figure 13-17. Examples of accounting output

The figure includes three examples of accounting information that the `amqsmon` sample program generates.

The first example shows information about the queue manager. The second example shows information about the objects that the queue manager controls, such as a queue. The third example shows more information about the queue.

Example of statistics data

```
MonitoringType: MQIStatistics
QueueManager: 'QM01'
IntervalStartDate: '2016-10-27'
IntervalStartTime: '11.41.38'
IntervalEndDate: '2016-10-27'
IntervalEndTime: '11.44.38'
CommandLevel: 900
ConnCount: 0
ConnFailCount: 0
ConnHighwater: 23
DiscCount: [0, 0, 0]
OpenCount: [0, 753, 0, 0, 0, 127, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 753, 0, 0, 0, 127, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [0, 829, 0, 0, 0, 204, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [829, 2]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [671324, 316]
GetCount: [829, 1]
. . .
```

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-18. Example of statistics data

Subscribing to statistics with the `amqsrua` program

Syntax for interactive mode: `amqsrua [-n MaxPubs] -m QmgrName`

- `MaxPubs` specifies number of reports to return before the command ends
- Enter the class (`-c`) of data to return
 - **CPU** returns information about CPU usage
 - **DISK** returns information about disk usage
 - **STATMQI** returns information about MQI usage
 - **STATQ** returns information about per-queue MQI usage
- Enter the type (`-t`) of data to return
 - For **CPU**: SystemSummary or QMgrSummary
 - For **DISK**: SystemSummary, QMgrSummary, or Log
 - For **STATMQI**: CONNDISC, OPENCLOSE, PUT, GET, SYNCPOINT, SUBSCRIBE, and PUBLISH
 - For **STATQ**: OPENCLOSE, INQSET, PUT, and GET

Figure 13-19. Subscribing to statistics with the `amqsrua` program

The `amqsrua` command reports metadata that is published by queue managers. This data can include information about the CPU, memory, and disk usage. You can also see data equivalent to the STATMQI PCF statistics data. The data is published every 10 seconds and is reported while the command runs.

The class of data that you select (CPU, DISK, STATMQI, or STATQ) determines the type of data available to return.

Example: Using the `amqsrua` sample program

```
$ amqsrua -n 2 -m QMA
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics

Enter Class selection
==> CPU
SystemSummary : CPU performance - platform wide
QMngrSummary : CPU performance - running queue manager

Enter Type selection
==> QMngrSummary
Publication received PutDate:20151014 PutTime:09175398
User CPU time - percentage estimate for queue manager 0.02%
System CPU time - percentage estimate for queue manager 0.04%
RAM total bytes - estimate for queue manager 200MB

Publication received PutDate:20151014 PutTime:09180405
User CPU time - percentage estimate for queue manager 0.00%
System CPU time - percentage estimate for queue manager 0.00%
RAM total bytes - estimate for queue manager 200MB
```

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-20. Example: Using the `amqsrua` sample program

The example shows the result of using `amqsrua` to view CPU performance data for the running queue manager over a 20-second period.

Real-time monitoring

- Determine the current state of queues and channels within a queue manager
 - Help to understand the steady state of the IBM MQ system
 - Determine condition of a queue manager at any moment, even if no specific event or problem was detected
 - Assist with determining the cause of a problem in the IBM MQ system
- Configure by using MQSC commands or IBM MQ Explorer queue manager properties
 1. Enable real-time monitoring at the queue manager
 2. Enable or disable real-time monitoring for individual queues or channels, or for multiple queues or channels

Real-time monitoring might negatively affect performance

With real-time monitoring you can determine the current state of queues and channels within a queue manager. The information that is returned is accurate at the moment the command is sent.

Information can be returned for one or more queues or channels and can vary in quantity.

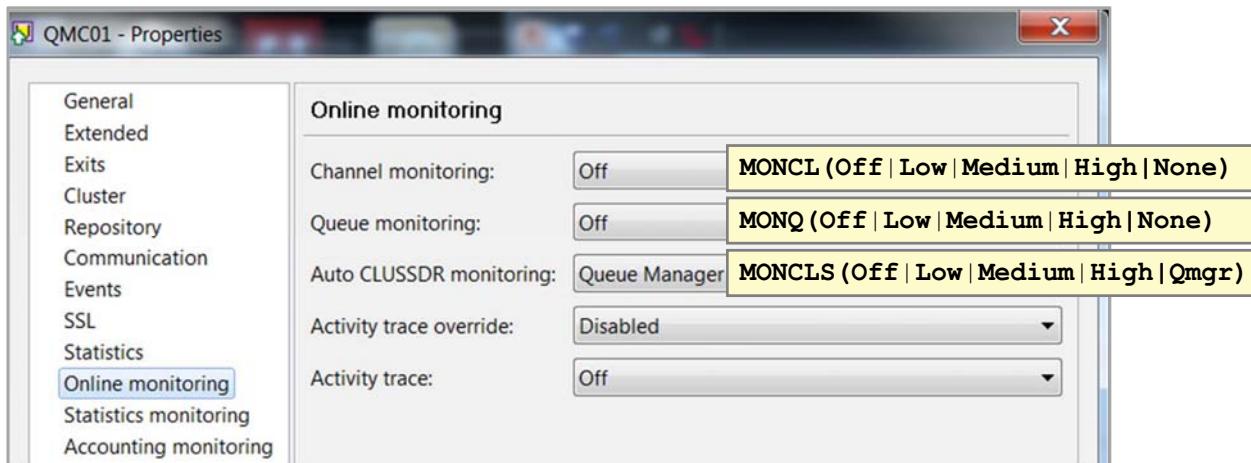
Real-time monitoring can be used in the following tasks:

- Helping system administrators understand the steady state of their MQ system to help with problem diagnosis when a problem occurs in the system.
- Determining the condition of your queue manager at any moment, even if no specific event or problem was detected.
- Assisting with determining the cause of a problem in your system.

With real-time monitoring, information can be returned for either queues or channels. Queue manager, queue, and channel attributes control the amount of real-time information that is returned.



Controlling queue manager real-time monitoring



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-22. Controlling queue manager real-time monitoring

The **Online monitoring** queue manager properties specify whether to collect real-time monitoring data about the current performance of channels that the queue manager hosts.

- To disable real-time monitoring data collection for the queue manager's channels that have the value **Queue Manager** in their **Channel monitoring** attribute, select **Off**.
- To disable online monitoring data collection for all queue manager channels regardless of the setting of the channel's **Channel monitoring** attribute, select **None**.

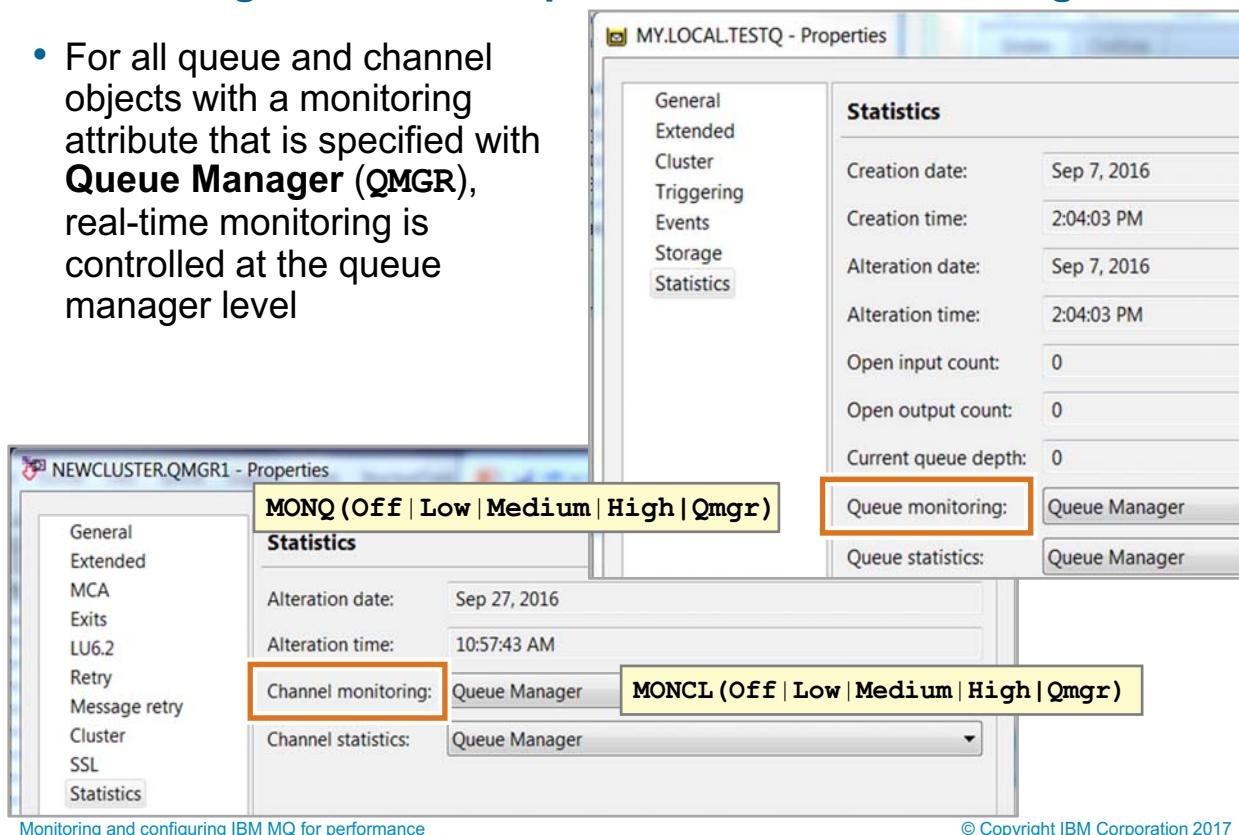
The **Queue monitoring** property specifies whether to collect online monitoring data about the current performance of queues that the queue manager hosts.

The **Auto CLUSSDR monitoring** property specifies whether to collect online monitoring data about the current performance of automatically defined cluster-sender channels.

The **Activity trace** property specifies whether to enable an activity trace. Activity trace was described in detail in Unit 9.

Controlling channel and queue real-time monitoring

- For all queue and channel objects with a monitoring attribute that is specified with **Queue Manager (QMGR)**, real-time monitoring is controlled at the queue manager level



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-23. Controlling channel and queue real-time monitoring

You can enable or disable real-time monitoring for individual queues or channels, or for multiple queues or channels.

To control individual queues, set the queue attribute `MONQ` to enable or disable real-time monitoring. To control many queues, enable or disable real-time monitoring at the queue manager level by using the queue manager attribute `MONQ`. For all queue objects with a monitoring attribute that is specified with the default value `QMGR`, real-time monitoring is controlled at the queue manager level.

To control individual channels, set the channel attribute `MONCHL` to enable or disable real-time monitoring. To control many channels, enable or disable real-time monitoring at the queue manager level by using the queue manager attribute `MONCHL`. For all channel objects with a monitoring attribute that is specified with the default value `QMGR`, real-time monitoring is controlled at the queue manager level.

Displaying queue and channel monitoring data

- Use IBM MQ Explorer content views or MQSC **DISPLAY QSTATUS** and **DISPLAY CHSTATUS** with **MONITOR** option

Example: On queue manager, MONCHL = MEDIUM

On sender channel QM1.QM2, MONCHL = QMGR

DISPLAY CHSTATUS (QM1.QM2) MONITOR

```
CHSTATUS (QM1. QM2)
XMITQ (Q1)
CONNNAME (127.0.0.1)
CURRENT
CHLTYPE (SDR)
STATUS (RUNNING)
SUBSTATE (MQGET)
MONCHL (MEDIUM)
XQTIME (755394737, 755199260)
NETTIME (13372, 13372)
EXITTIME (0, 0)
XBATCHSZ (50, 50)
COMPTIME (0, 0)
STOPREQ (NO)
RQMNAME (QM2)
```

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-24. Displaying queue and channel monitoring data

To display real-time monitoring information for a queue or channel, use either the MQ Explorer or the appropriate MQSC command. Some monitoring fields display a comma-separated pair of indicator values, which help you to monitor the operation of your queue manager.

Configuring and tuning IBM MQ for performance

- Default properties are configured to produce a fully functioning queue manager by using reasonable amounts of memory and disk space
 - For on-premises installations, IBM MQ is not optimized for performance
 - For IBM MQ Appliance, IBM MQ is optimized for performance
- Apply tuning to all connected queue managers because messaging performance by using more than one queue manager depends on the performance of those other queue managers
- Tuning options:
 - Queue manager logs
 - Queue manager channels
 - Queue manager listeners

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-25. Configuring and tuning IBM MQ for performance

Unless the MQ queue manager runs on an MQ Appliance, an MQ queue manager with default properties is configured to produce a fully functioning queue manager that uses reasonable amounts of memory and disk space. It is not optimized for performance.

You can make some configuration changes to improve the performance of message processing with MQ.



Attention

It might not be necessary to implement the tuning guidelines that are described in this unit, especially if the message throughput and response time of the queue manager already meet the required level. If applied inappropriately, implementation of some of the tuning options can degrade the performance of a previously balanced system.

Carefully monitor the results of tuning the queue manager to ensure that no adverse effects are evident.

Always thoroughly test your environment after you change any tuning parameters.

Queue manager logs

- Might require performance tuning when the queue manager processes persistent messages, which are stored in logs
- Default settings for new queue managers:
 - **Default log settings** in IBM MQ Explorer properties
 - **LogDefaults** stanza in the `mqsc.ini` file
 - Can override default settings by using `qm.ini` or queue manager **Properties** in IBM MQ Explorer
- Performance factors:
 - Log type*
 - Log file path*
 - Log file pages*
 - Level log write integrity
 - Log buffer pages
 - Number of primary and secondary log files
 - Number of concurrent applications
 - Application processing within a unit of work

* Must be specified when the queue manager is created and cannot be changed

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

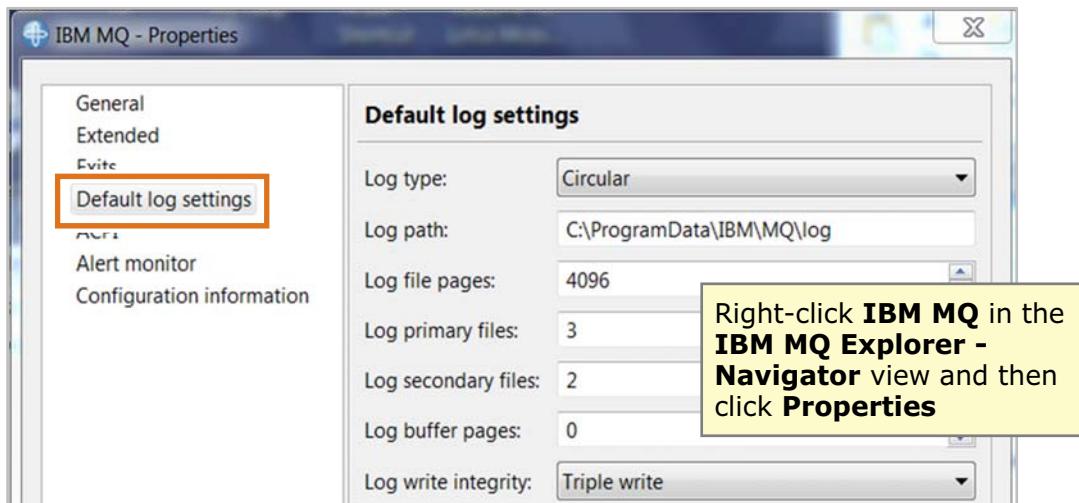
Figure 13-26. Queue manager logs

The location, size, and number of queue manager logs can affect performance. Some performance factors, such as log file location and the type of logging, must be specified when the queue manager is created. Other performance factors such as level of log integrity, log file pages, and log buffer pages can be changed at any time but might require a restart.

Queue manager log settings can be configured in the MQ Explorer queue manager properties and the **LogDefaults** stanza in the `mqsc.ini` file.

Default log settings in IBM MQ Explorer

- Specifies log settings to use for new queue managers that are created in IBM MQ Explorer



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-27. Default log settings in IBM MQ Explorer

The queue manager log properties that can be configured in the MQ Explorer are:

- Log type
- Log path
- Log file pages
- Log primary files
- Log secondary files
- Log buffer pages
- Log write integrity

To access the **Default log settings** in MQ Explorer:

- Right-click the queue manager in the **MQ Explorer - Navigator** view and then click **Properties**.
- From the **Properties** list, click **Default log settings**.

Log type

- Linear logging file extents are continually allocated as required
- Circular logging log file extents are reused after they no longer contain active log data
- For performance, choose circular logging if linear logging is not required for re-creating lost or damaged data by replaying the contents of the log
- Type of logging cannot be changed after a queue manager is created

Figure 13-28. Log type

The type of logging can affect performance.

Circular logging is more efficient because log files are reused when they no longer contain active log data. If circular logging is enabled, ensure that **Log primary files** are greater than 20. Ensure that primary logs satisfy circular logging because secondary logs are formatted each time that they are reused.

Log path

- Specifies directory for the queue manager log files
- Locate the queue manager log on its own disk, particularly when processing large messages, or high message volumes (> 50 messages per second)
 - On Windows, create a directory on fastest local disk available and then specify the directory by using the `-1d` attribute when creating the queue manager
 - On UNIX and Linux, allocate and mount a file system for queue manager files and log before creating the queue manager
- When possible, allocate the log on a device with a battery-backed write cache or use fastest local disk available

Figure 13-29. Log path

The log file path and location can affect performance.

Nonpersistent messages are held in main memory and then saved to the file system as the queue depth increases. Persistent messages are synchronously written to the log.

Queue and log I/O contention can occur due to the queue manager simultaneously updating a queue file and log extent on the same disk.

To avoid potential queue and log I/O contention, put queues and logs on separate and dedicated physical devices.

Log file pages

- Specified on queue manager creation and cannot be changed
- Defines the size of one physical disk extent in units of 4 KB pages
- For Windows:
 - Default number of log file pages is 4096, giving a log file size of 16 MB
 - Minimum number of log file pages is 32
 - Maximum number of log file pages is 65536
- For UNIX and Linux:
 - Default number of log file pages is 4096, giving a log file size of 16 MB
 - Minimum number of log file pages is 64
 - Maximum number of log file pages is 65,535
- For performance, allocate maximum size if disk space is available

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-30. Log file pages

The **Log file pages** option in the queue manager log properties defines the size of one physical disk extent.

The size of the disk extent is directly related to the elapsed times between changing disk extents. It is better to have a smaller number of large extents. The largest size reduces the frequency of switching extents.

Log file page size can be configured in the MQ Explorer queue manager properties.

Log primary and secondary files

- Queue manager allocates and formats primary log file extents when it is first started or when extra extents are added
 - Minimum = 2
 - Maximum = 254 on Windows, or 510 on UNIX and Linux
 - Default = 3
- Queue manager dynamically allocates secondary log file extents when the primary files are exhausted
 - Minimum = 1
 - Maximum = 253 on Windows, or 509 on UNIX and Linux
 - Default = 2
- Total number of primary and secondary log files:
 - Must not exceed 255 on Windows, or 511 on UNIX
 - Must not be less than 3
- Change in value requires queue manager restart
- For performance, ensure that a reasonable number of secondary extents exist for system activity

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-31. Log primary and secondary files

Records are written into the log buffer with each put, get, and commit. This information is moved onto the log disk. Periodically the checkpoint process decides how many of these log file extents that are in the *active* log must be kept online for recovery purposes. The log extents no longer in the active log are available for achieving when the queue manager uses linear logs.

Ensure that sufficient primary logs are available to hold the active log plus the new log extents that are used until the next checkpoint; otherwise, some secondary logs are temporarily included in the log set. Secondary logs must be instantly formatted, which is an unnecessary delay when the queue manager uses circular logging.

The value for the number of logs is examined when the queue manager is started. You can change this value, but changes do not become effective until the queue manager is restarted, and even then the effect might not be immediate.

Log buffer pages

- Log buffer is a circular piece of main memory where the log records are concatenated so that multiple log records can be written to the log file in a single I/O operation
- Specify the size of the buffers in units of 4 KB pages
 - Minimum = 18
 - Maximum = 512
 - If you specify 0, queue manager selects size
- Improve persistent message throughput of large messages (message size > 1 MB) by increasing **Log buffer pages** to improve likelihood of messages that need only one I/O to get to the disk
- Requires queue manager restart to recognize any changes

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-32. Log buffer pages

To improve persistent message throughput of large messages, increase the value in the **Log buffer pages** property to improve the likelihood of messages that need only one I/O to get to the disk. A large message is a message greater than 1 MB.

You can reduce the memory by using a smaller log buffer page without affecting throughput in environments that process a few (under 100) small persistent messages. A small message is a message less than 10 KB.

Log write integrity level

- Method that the queue manager logger uses to reliably write log records
 - **SingleWrite**: Safe for the logger to write log records in a single write because the hardware assures full write integrity
 - **TripleWrite**: Default method that assures full write integrity when write integrity hardware is not available
- On Windows, change **LogWriteIntegrity** string value under `IBMMQSeries\CurrentVersion\Configuration\QueueManager` in the Windows registry
- On UNIX and Linux, change **LogWriteIntegrity** value in **Log** stanza of `qm.ini` file
- Level change requires queue manager restart

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-33. Log write integrity level

MQ provides full write integrity of logs where hardware that assures write integrity is not available.

Some hardware promises if a write operation writes a page and fails for any reason, a subsequent read of the same page shows that each byte in the buffer is the same as before the write.

On this type of hardware, it is safe for the logger to write log records in a single write as the hardware assures full write integrity. This method provides the highest level of performance.

The log write integrity level can be configured in the MQ Explorer queue manager properties and in the **Log** stanza of the queue manager configuration file.

Queue manager Log settings in qm.ini

```

#* Module Name: qm.ini                                     *#
#* Type : IBM MQ queue manager configuration file        *#
# Function : Define the configuration of a single queue manager *#
#*                                                       *#
#*****                                                       *#
#* Notes :                                                 *#
#* 1) This file defines the configuration of the queue manager *#
#*                                                       *#
#*****                                                       *#
ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

. . .

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
 LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0 1
  LogWriteIntegrity = TripleWrite
  LogDefaultPath=/var/mqm/log/saturn!queue!manager/

```

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-34. Queue manager Log settings in qm.ini

This figure shows an example of the **Log** stanza in the **qm.ini** file.

Queue manager channels performance tuning

- If environment is stable, run channels as trusted or fast path IBM MQ applications to give a performance benefit through reduced code pathlength
- Configure by using one of two options:
 - Specify a value of **MQIBindType=FASTPATH** in the **Channels** stanza of the **qm.ini** or registry file
 - Set the environment variable **MQ_CONNECT_TYPE = FASTPATH** in the environment in which the channel is started
- Do not use trusted or fast path channels:
 - If channel exits are used, a potential exists for the exit to corrupt the queue manager if the exits are not correctly written and thoroughly tested
 - If **STOP CHANNEL MODE (FORCE)** command is used
 - If the environment is unstable with regular component failure

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-35. Queue manager channels performance tuning

If your MQ environment is stable, fast path channels can increase throughput for both nonpersistent and persistent messaging. For persistent messages, the improvement is for the path through the queue manager, and does not affect performance when writes to the log disk occur.



Note

Because the greater proportion of time for persistent messages is while writing to the log disk, the performance improvement for fast path channels is less apparent with persistent messages than with nonpersistent messages.

Setting the **MQIBindType** attribute in the **qm.ini** file to FASTPATH causes the channel to run in “trusted” mode. Trusted applications do not use a thread in the agent process. No interprocess communication (IPC) between the channel and agent exists because the agent does not exist in this connection.

If the channel is run in STANDARD mode, then any messages that are passed between the channel and agent use IPC memory that is dynamically obtained and held only for the lifetime of the MQGET. Standard channels each require an extra 80 KB of memory. When the message rate increases, more IPC memory is used in parallel.

Queue manager listeners performance tuning

- Run as trusted or fast path IBM MQ applications to give a performance benefit through reduced code pathlength
- Set the environment variable `MQ_CONNECT_TYPE=FASTPATH` in the environment in which the listener is started

- Works for listeners that are started when the `runmqlsr` command is run manually or in a script
 - `MQ_CONNECT_TYPE=FASTPATH` needs to be present only in the shell from which the `runmqlsr` command is entered

- Works for listeners that were defined by using the `DEFINE LISTENER` MQSC command
 - `MQ_CONNECT_TYPE=FASTPATH` must be set in the environment in which the queue manager is started

[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-36. Queue manager listeners performance tuning

Running queue manager listeners as trusted or fast path applications might give a performance benefit through reduced code path length.

More resource savings are available by using the `runmqlsr` listener command rather than InetD, including a reduced requirement on virtual memory, number of processes, and file handles.



Note

The performance improvement for fast path channels is less apparent with persistent messages than with nonpersistent messages.

Unit summary

- Describe the statistics and accounting data that IBM MQ provides
- View and generate accounting and statistical data
- Subscribe to IBM MQ statistic topics
- Interpret statistics and accounting data to identify possible system performance benefits
- Configure and tune IBM MQ for improved performance

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-37. Unit summary

Review questions

1. Which of the following are types of statistics that are collectable by IBM MQ?
 - A. Queue
 - B. Process
 - C. Channel
 - D. Queue manager

2. Select all answers that are correct for the following statement.
Accounting data includes:
 - A. Application data
 - B. Queue data
 - C. Publish/subscribe data
 - D. Byte counts
 - E. All of the above



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-38. Review questions

Write your answers down here:

- 1.

- 2.

Review answers

1. Which of the following are types of statistics that are collectable by IBM MQ?
 - A. Queue
 - B. Process
 - C. Channel
 - D. Queue manager

The answer is A, C, and D.
2. Select all answers that are correct for the following statement.
Accounting data includes:
 - A. Application data
 - B. Queue data
 - C. Publish/subscribe data
 - D. Byte counts
 - E. All of the above

The answer is E.



Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-39. Checkpoint answers

Exercise: Monitoring IBM MQ for performance

Monitoring and configuring IBM MQ for performance

© Copyright IBM Corporation 2017

Figure 13-40. Exercise: Monitoring IBM MQ for performance

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

Exercise objectives

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



[Monitoring and configuring IBM MQ for performance](#)

© Copyright IBM Corporation 2017

Figure 13-41. Exercise objectives

See the *Course Exercises Guide* for detailed instruction.